1998-12

A simulation of MPSK communications system performance in the presence of wideband noise and co-channel interference

Erdogan, Veysel

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A SIMULATION OF MPSK COMMUNICATIONS SYSTEM PERFORMANCE IN THE PRESENCE OF WIDEBAND NOISE AND CO-CHANNEL INTERFERENCE

by

Veysel Erdogan

December 1998

Thesis Advisor: Jovan Lebaric
Thesis Co-Advisor: Clark Robertson

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 1998	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE A SIMULATION OF MPSK COMMUNICATIONS SYSTEM PERFORMANCE IN THE PRESENCE OF WIDEBAND NOISE AND CO-CHANNEL INTERFERENCE	**5. FUNDING NUMBERS**
6. AUTHOR(S) Erdogan, Veysel	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

The purpose of this research is to model digital communications systems in the time domain using MATLAB Simulink and the Communications Toolbox and to determine and verify the system performance in the presence of additive noise and interference.While the theoretical results are available for the assesment of the influence of wideband gaussian noise on the performance of a digital communications system,determining the performance of a system in the presence of noise and interference requires a computer simulation.The time domain modeling allows the visualisation of communication signals at various stages of transmitters and receivers and "Monte Carlo" type simulations to establish the bit error rates under realistic conditions (noise, interference) and for different values of transmitter/receiver parameters, different channel parameters and different types of interfering signals. It should be noted that the simulation does not account for system non-linearities.

14. SUBJECT TERMS Simulink,Matlab, Noise and Interference, Probability of Bit Error	**15. NUMBER OF PAGES** 118
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std.

Approved for public release; distribution is unlimited

A SIMULATION OF MPSK COMMUNICATIONS SYSTEM PERFORMANCE IN THE PRESENCE OF WIDEBAND NOISE AND CO-CHANNEL INTERFERENCE

Veysel Erdogan
First Lieutenant, Turkish Army
B.S., Turkish Military Academy, 1991

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1998

Author: [redacted]
Veysel Erdogan

Approved by: [redacted]

Clark Robertson, Thesis Co-Advisor

[redacted]

Jeffrey B. Knorr, Chairman
Department of Electrical Engineering

ABSTRACT

The purpose of this thesis is to model digital communications systems in the time domain using MATLAB Simulink and the Communications Toolbox as well as to determine and verify system performance in the presence of additive noise and co-channel interference. While the theoretical results are available for the effect of wideband gaussian noise on the performance of digital communications systems, determining the performance of a system in the presence of noise and co-channel interference is best done by computer simulation. Time domain modeling allows the visualization of the communication signal at various stages, and "Monte Carlo" type simulations establish the bit error rates under realistic conditions of noise and co-channel interference as well as for different transmitter/receiver parameters, different channel parameters, and different types of interfering signals. It should be noted that the simulation does not account for system non-linearities.

TABLE OF CONTENTS

I. INTRODUCTION 1

II. SIMULINK AND THE COMMUNICATIONS TOOLBOX 3

III. PHASE-SHIFT-KEYING 5

A. BINARY PHASE-SHIFT-KEYING 5

B. SIGNAL CONSTELLATION FOR BPSK 6

C. PROBABILITY OF BIT ERROR FOR BPSK 7

D. QUADRATURE PHASE-SHIFT-KEYING 10

E. SIGNAL CONSTELLATION FOR QPSK 11

F. PROBABILITY OF BIT ERROR FOR QPSK 12

G. SIMULINK MODEL OF MPSK DIGITAL COMMUNICATION SYSTEM 12

IV. BIT ERROR PROBABILITY CONVERGENCE 17

V. SIMULATION VERIFICATION FOR BPSK AND QPSK WITH AWGN 31

A. BPSK BIT ERROR PROBABILITY ESTIMATES 33

1.BPSK Results for the Simulink Model 33

2.BPSK Results Obtained by Using Matlab Program MPSK1.M 35

B. QPSK BIT ERROR PROBABILITY ESTIMATES 38

1.QPSK Results for the Simulink Model 38

2.QPSK Results Obtained by Using Matlab Program MPSK1.M 39

VI. SIMULATION VERIFICATION FOR BPSK AND QPSK WITH ADDITIVE WHITE GAUSSIAN NOISE AND CO-CHANNEL INTERFERENCE 41

A. BPSK BIT ERROR PROBABILITY ESTIMATES 41

1.BPSK Results for the Simulink Model 43

2.BPSK Results Obtained by Using Matlab Program MPSK1.M 46

B. QPSK BIT ERROR PROBABILITY ESTIMATES 49

1.QPSK Results for the Simulink Model 49

2.QPSK Results Obtained by Using Matlab Program MPSK1.M 52

VII. CONCLUSIONS AND RECOMMENDATIONS 57

A. CONCLUSIONS 57

B. RECOMMENDATIONS 58

APPENDIX A. BATCH_RUN.M MATLAB PROGRAM FOR CONVERGENCE TEST OF SIMULATION USING BPSK 59

APPENDIX B. BER_PLOT.M MATLAB PROGRAM FOR PLOTTING THE FIGURES OF THE SIMULATION FOR EACH VARIANCES 61

APPENDIX C. PLOT_SIM_NOISE2 MATCHAD PROGRAM FOR RESULTS OF NOISE ONLY CASE BPSK USING SIMULINK MODEL 63

APPENDIX D. PLOT_MAT_NOISE2 MATCHAD PROGRAM FOR RESULTS OF NOISE ONLY CASE BPSK USING MATLAB PROGRAM 65

APPENDIX E. PLOT_SIM_NOISE4 MATCHAD PROGRAM FOR RESULTS OF NOISE ONLY CASE QPSK USING SIMULINK MODEL 67

APPENDIX F. PLOT_MAT_NOISE4 MATCHAD PROGRAM FOR RESULTS OF NOISE ONLY CASE QPSK USING MATLAB PROGRAM 71

APPENDIX G. PLOT_SIM_NOISE&INT2 MATCHAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE BPSK USING SIMULINK MODEL 75

APPENDIX H. PLOT_MAT_NOISE&INT2 MATCHAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE BPSK USING MATLAB PROGRAM 79

APPENDIX I. PLOT_SIM_NOISE&INT4 MATCHAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE QPSK USING SIMULINK MODEL 83

APPENDIX J. PLOT_MAT_NOISE&INT4 MATCHAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE QPSK USING MATLAB PROGRAM 87

APPENDIX K. PREPARE.M MATLAB PROGRAM TO ENTER THE SYSTEM PARAMETERS TO BE USED BY THE SIMULINK MODEL 91

APPENDIX L. MPSK.M MATLAB PROGRAM TO START SIMULATION 93

APPENDIX M. MPSK1.M MATLAB PROGRAM FOR USING MATLAB PROGRAM OPTION VICE SIMULINK MODEL 95

LIST OF REFERENCES 101

BIBLIOGRAPHY 103

INITIAL DISTRIBUTION LIST 105

ACKNOWLEDGMENT

I would like to thank all that contributed to and helped me prepare this thesis: Dr. Jovan Lebaric for his support and advice and Prof. Clark Robertson whose knowledge in Communications helped a lot in my studies.

Second I would like to thank my family in Turkey, for their constant support and encouragement to accomplish this hard work.

Lastly I want to thank my country and Turkish Armed Forces for sending me to NPS to have this education and opportunity. NPS not only gave me a good technical knowledge but also awarded me a different way of looking at life.

I. INTRODUCTION

A military site or a platform typically has a large number of communication systems operating simultaneously in both transmit and receive modes. The interference of one system with another depends on available frequency allocations, the physical separation of transmit and receive antennas, non-linearity of transmitter output stages, and so on. The objective of this study is to obtain a computer-based prediction of noise and co-channel interference effects on the probability of bit error for M-ary Phase Shift Keying (MPSK) digital communication systems. To accomplish this, a model was developed using SIMULINK and the Communications Toolbox. The performance of the model has been verified by comparing the simulation results and the theoretical results for the bit error probability in the presence of additive white Gaussian noise (AWGN). The model has been subsequently modified to include co-channel interference, and the bit error probability of MPSK with AWGN and co-channel interference was obtained via simulation.

II. SIMULINK AND THE COMMUNICATIONS TOOLBOX

SIMULINK is a program for modeling linear and non-linear dynamic systems in the time-domain. Models in SIMULINK are represented in block-diagram form and can be assembled from block libraries and sub-libraries. Furthermore, SIMULINK allows the display of results and changes of certain model parameters without interrupting the simulation. Since SIMULINK is built upon the MATLAB numeric computation system, it offers direct access to the MATLAB workspace and MATLAB's mathematical and engineering functions. SIMULINK's main features are:

- modeling and analysis of dynamic systems, including linear, nonlinear, continuous, discrete, and hybrid
- flexible "open system" environment that allows addition of new blocks to Simulink
- seamless interface with MATLAB's built-in math functions, 2-D and 3-D graphics, and add-on toolboxes for specialized applications
- choice of methods of running a simulation (menu-driven on-screen or batch-mode)
- an optimized computer platform implementation that ensures fast and accurate results
- unlimited model size.

A typical SIMULINK session starts by either defining a new model or recalling a previously defined model and then proceeds to a simulation of the performance of that model. In practice these two steps are often performed iteratively as the model designer creates and modifies a model to achieve the desired behavior [Ref.1].

The Communications Toolbox is a collection of MATLAB functions and SIMULINK blocks for model development and simulation in the communications area. The functions/blocks are organized in the following sub-categories: Data Source, Source Coding and Decoding, Error-Control Coding, Modulation and Demodulation, Transmission and Reception filters, Transmitting Channel, Multiple Access, and Synchronization and Utilities [Ref.1].

III. PHASE SHIFT KEYING

Modulation of a sinusoidal carrier allows the conveyance of information in binary form. The baseband information bit stream modulates the carrier by means of discrete changes in the carrier amplitude (amplitude-shift keying or ASK) carrier frequency (frequency-shift keying or FSK) or carrier phase (phase-shift keying or PSK) or some combination of the above. Each different state of the carrier, known as a symbol, corresponds to one or more bits of baseband information [Ref 2.].

A. BINARY PHASE-SHIFT KEYING

In binary phase-shift keying (BPSK), the phase of a constant amplitude carrier signal is switched between two values according to the two possible "messages" m_1 and m_2 corresponding to binary 1 and 0, respectively. Normally, the two phases are separated by 180 degrees. If the sinusoidal carrier has amplitude A_c, then the average energy per bit $E_b = \frac{1}{2}A_c^2 T_b$, and the transmitted BPSK signal can be represented as:

$$S_{BPSK}(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \Theta_c) \qquad 0 \le t \le T_b \qquad (\text{ binary 1}) \tag{1}$$

$$S_{BPSK}(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi + \Theta_c) \qquad 0 \le t \le T_b \qquad (\text{ binary 0}) \tag{2}$$

$$S_{BPSK}(t) = -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \Theta_c) \qquad 0 \le t \le T_b \qquad (\text{ binary 0}) \tag{3}$$

where E_b is the average energy per bit, T_b is the bit duration, and a rectangular pulse shape p(t)=$\pi[(t-T_b/2)/T_b)]$ is assumed. For this signal set, the single waveform

$$\phi_1(t)=\sqrt{\frac{2}{T_b}}\cos(2\pi f_c t) \qquad 0 \le t \le T_b \tag{4}$$

can be defined. Using this signal, the BPSK signal set can be represented as

$$S_{BPSK}=\{\sqrt{E_b}\phi_1(t), -\sqrt{E_b}\phi_1(t)\} \tag{5}$$

B. SIGNAL CONSTELLATION FOR BPSK

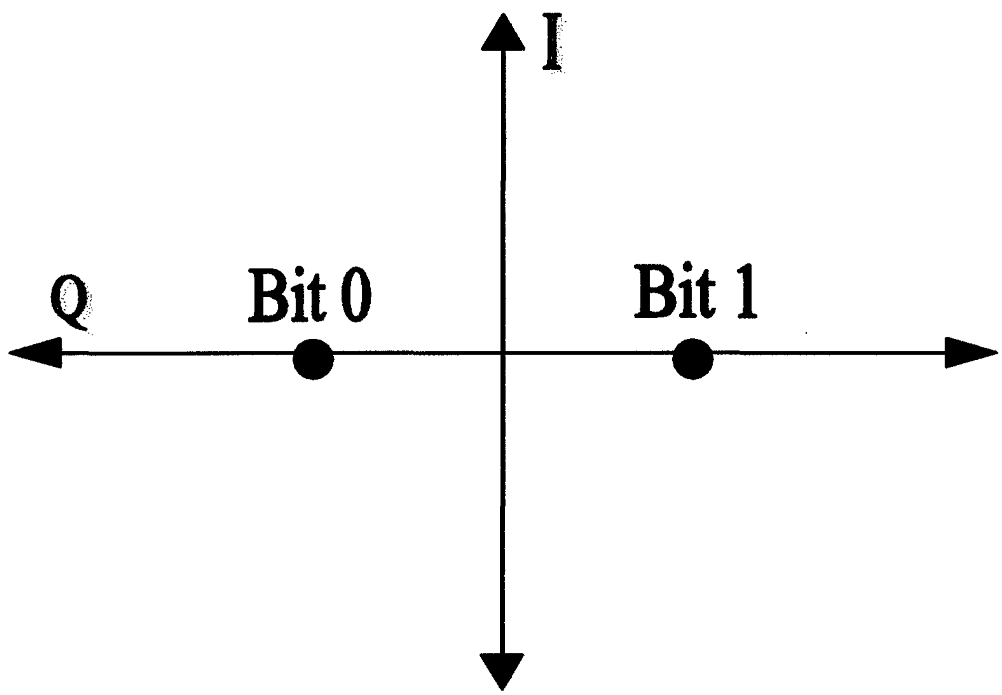


Figure 1. Signal Constellation for BPSK

A BPSK signal can be geometrically represented in " signal space" as shown below. Such a representation is referred to as the symbol constellation and provides a graphical representation of the complex envelope of a BPSK signal for the two possible symbols. The x-axis of the constellation diagram represents the "in-phase" component of the complex envelope, and the y-axis represents the "quadrature" component of the complex envelope. The distance between the signals on a constellation diagram indicates how different the modulation waveforms are and determines how well a receiver can differentiate between all possible symbols when random noise is present. The larger the signal distance, the better the chance of correct symbol detection.

C. PROBABILITY OF BIT ERROR FOR BPSK

An important measure of performance used for digital modulation is the probability of symbol error, P_E. It is often convenient to specify system performance by the probability of bit error P_B, even when decisions are made on the basis of symbols rather than bits. The relationship between P_B and P_E for orthogonal signaling is:

$$P_B = \frac{P_E}{\log_2 M} \tag{6}$$

For the BPSK modulation (M=2), the symbol error probability is equal to the bit error probability. When the signals are assumed equally likely and signal $s_i(t)$ (i=1,2) is transmitted, the received signal, $r(t)$, is equal to $s_i(t)+n(t)$, where $n(t)$ is modeled as additive white Gaussian noise (AWGN). The antipodal signals (signals of equal amplitude and opposite polarity) $s_1(t)$ and $s_2(t)$ are (cf. Eq.5):

$$s_1(t) = \sqrt{E_b}\,\phi_1(t) \qquad 0 \le t \le T_b \tag{7}$$

$$s_2(t) = -\sqrt{E_b}\,\phi_1(t) \qquad 0 \le t \le T_b \tag{8}$$

The decision stage of the detector will choose the $s_i(t)$ with the largest correlator output $z_i(t)$, or in this case of equal-energy antipodal signals, the detector, using the decision rule, decides:

$$s_1(t) \quad \text{if} \quad z(T) > \gamma_0$$

$$s_2(t) \quad \text{otherwise}$$

where γ_0 denotes the decision threshold (equal to 0 for equally probable antipodal signals) and $z(T)$ is the correlator output at time T. Two types of detection error can be made. The first type of error takes place if $s_1(t)$ is transmitted but the noise is such that the detector measures a negative value for $z(T)$ and decides (incorrectly) that signal $s_2(t)$ was sent. The second type of error takes place if signal $s_2(t)$ is transmitted but the detector measures a positive value for $z(T)$ and decides (again incorrectly) that signal $s_1(t)$ was sent. Therefore, the probability of bit error, P_B, is the sum of two conditional probabilities:

$$P_B = P(s_2 \mid s_1)P(s_1) + P(s_1 \mid s_2)P(s_2) \tag{9}$$

For the case when the symbol probabilities are known a priori and the symbols are equally probable (which is mostly the case):

$$P(s_1) = P(s_2) = \frac{1}{2} \tag{10}$$

The expression for the bit error probability becomes:

$$P_B = \frac{1}{2}P(s_2 \mid s_1) + \frac{1}{2}P(s_1 \mid s_2) \tag{11}$$

Because of the symmetry of the probability density functions (pdf's) of the sum of signal and noise, $P(z \mid s_1)$ and $P(z \mid s_2)$ the two conditional probabilities are also identical and

$$P_B = P(s_2 \mid s_1) = P(s_1 \mid s_2) \tag{12}$$

The probability of a BPSK bit error P_B is numerically equal to the area under the "tail" of either pdf $p(z \mid s_1)$ or $p(z \mid s_2)$ that falls on the incorrect side of the threshold. We can therefore compute P_B by either integrating $p(z \mid s_1)$ between the limits $-\infty$ and γ_0 or by integrating $p(z \mid s_2)$ between the limits γ_0 and ∞. Hence,

$$P_B = \int_{\gamma_0=(a_1+a_2)/2}^{\infty} p(z \mid s_2)dz \tag{13}$$

where the conditional pdf's $p(z \mid s_1)$ $(i = 1,2)$ are Gaussian with mean value a_i, variance σ_0, and the optimum threshold, γ_0, is $(a_1 + a_2)/2$. The probability of bit error for BPSK is:

$$P_B = \int_{\gamma_0=(a_1+a_2)/2}^{\infty} \frac{1}{\sigma_0\sqrt{2\pi}} \exp[-\frac{1}{2}(\frac{z-a_2}{\sigma_0})^2]dz \tag{14}$$

If we introduce $u = \dfrac{z-a_2}{\sigma_0}$ and $\sigma_0 du = dz$,

the integral simplifies to

$$P_B = \int_{u=(a_1-a_2)/2\sigma_0}^{u=\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{u^2}{2})du = Q(\frac{a_1-a_2}{2\sigma_0}) \tag{15}$$

where σ_0 is the standard deviation of the noise at the output of the correlator. The function $Q(x)$ is defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp(-\frac{u^2}{2})du \tag{16}$$

For equal energy antipodal signaling, such as the BPSK format in (5), the receiver output signal components are $a_1 = \sqrt{E_b}$ when $s_1(t)$ is sent and $a_2 = -\sqrt{E_b}$ when $s_2(t)$ is sent. For AWGN the noise variance $\sigma_0^{\,2}$ at the correlator output is equal to $N_0/2$ so that we can rewrite probability of bit error as follows [Ref.3]:

$$P_B = \int_{\sqrt{2E_b/N_0}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{u^2}{2}) du \tag{17}$$

$$P_B = Q(\sqrt{\frac{2E_b}{N_0}}) \tag{18}$$

D. QUADRATURE PHASE-SHIFT KEYING

In quadrature phase-shift keying (QPSK), two bits are transmitted in a single modulation symbol. The phase of the carrier takes on one of four equally spaced values such as 0, $\pi/2$, π, and $3\pi/2$, where each value of phase corresponds to a unique pair of message bits. A QPSK signal can be defined as

$$S_{QPSK}(t) = \sqrt{\frac{2E_s}{Ts}} \cos[2\pi f_c t + (i-1)\frac{\pi}{2}] \qquad 0 \le t \le T_s \quad i = 1,2,3,4. \tag{19}$$

where T_s denotes symbol duration (equal to twice the bit duration T_b). By using trigonometric identities, Equation 19 can be written in the interval $0 \le t \le T_s$ as:

$$S_{QPSK}(t) = \sqrt{\frac{2E_s}{Ts}} \cos[(i-1)\frac{\pi}{2}] \cos(2\pi f_c t) - \sqrt{\frac{2E_s}{Ts}} \sin[(i-1)\frac{\pi}{2}] \sin(2\pi f_c t) \tag{20}$$

If the basis functions $\Phi_1(t) = \sqrt{2/T_S}\cos(2\pi f_c t)$ and $\Phi_2(t) = \sqrt{2/T_S}\sin(2\pi f_c t)$ are defined over the interval $0 \le t \le T_s$ then the QPSK signal can be expressed in terms of the basis signals as [Ref.4]:

$$S_{QPSK}(t) = \{\sqrt{E_s}\cos[(i-1)\frac{\pi}{2}]\Phi_1(t) - \sqrt{E_s}\sin[(i-1)\frac{\pi}{2}]\Phi_2(t)\} \quad i = 1,2,3,4. \qquad (21)$$

E. SIGNAL CONSTELLATION FOR QPSK

The QPSK signal constellation is shown below. The constellation suggests that a QPSK signal can be thought of as a combination of two pairs of antipodal BPSK signals, with the two BPSK pairs orthogonal (in phase quadrature) to each other.

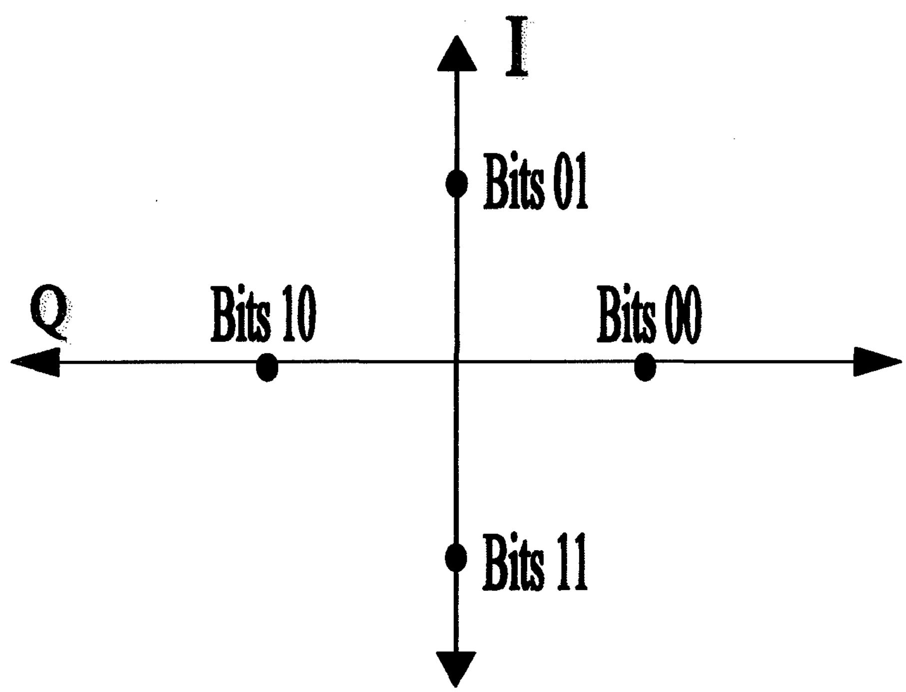


Figure 2. Signal Constellation for QPSK

F. PROBABILITY OF BIT ERROR FOR QPSK

Since QPSK may be considered as a combination of two orthogonal BPSK signals, the bit QPSK error probability is identical to the BPSK bit error probability given by Equation 18. Note however that the QPSK symbol error probability is given [Ref.4] as:

$$P_s = 2Q(\sqrt{\frac{2E_b}{N_0}})[1-\frac{1}{2}Q(\sqrt{\frac{2E_b}{N_0}})] \tag{22}$$

G. SIMULINK MODEL OF MPSK DIGITAL COMMUNICATION SYSTEM

In the simulation of communication systems, there are two options for representing the modulation/demodulation process: passband and baseband. In passband simulations, the carrier signal is included in the simulation model. By the Nyquist sampling theorem, the simulation sampling frequency must be at least twice the sum of the carrier signal frequency and the highest frequency component of the message in order to recover the message correctly. Since the frequency of the carrier is usually much greater than the highest frequency of the input message signal, a passband simulation requires a very large number of samples and excessive simulation times. The simulation time can be dramatically reduced by using baseband-equivalent simulation, also known as the low-pass equivalent method. A baseband equivalent model uses complex envelopes to reduce the simulation sampling rate and thus speed up the simulation. As the name suggests, the complex envelope is a complex function of time that carries information about both the time-varying amplitude and the time-varying phase of a passband signal. Since the amplitude of an MPSK signal does not vary with time, the complex envelope of an MPSK signal has a constant magnitude but a time-varying phase angle of the complex envelope for the *i*–th symbol is $2\pi i/M$ [Ref.1]. In this thesis the baseband equivalent model for MPSK, shown in Figure 3, was developed and applied to 2PSK and 4PSK cases.

In order to evaluate the effect of co-channel interference, the model in Figure 3 was modified by adding another MPSK modulator and another message source.

The MPSK model has the following main functional blocks:

- Sampled Read from Workspace
- MPSK modulation and MPSK demodulation blocks
- AWGN channel
- K-step Delay
- Error Counter
- Triggered Write to File.

In addition to these blocks, auxiliary blocks such as multiplexer, oscilloscope, sum, pulse generator, switch, counter, and so on are also used.

The *Sampled Read From Workspace* block reads a row of data for a workspace variable (a matrix) at every data sampling point. The output of this block is a vector whose length is equal to the number of columns of the workspace variable matrix. The possibility exists to offset (postpone) reading from workspace by a desired value such

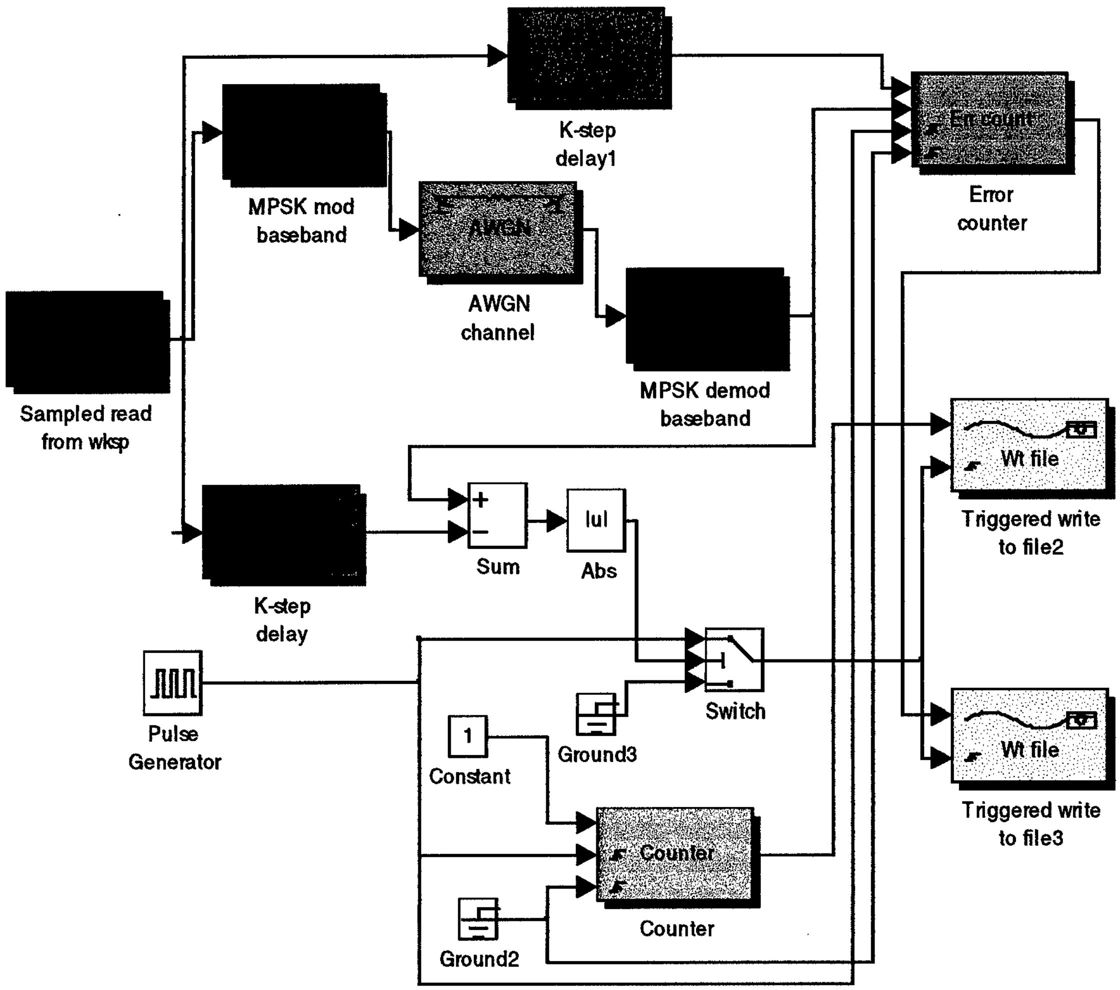


Figure 3. Model of MPSK Communication System

that up to the user-defined offset time the output of this block is equal to a user-defined initial value. When a non-zero time offset is used, reading from workspace starts once the simulation time exceeds the offset. In the case that the last row of the workspace matrix is reached before completing the simulation (that is when the number of rows in the workspace matrix is less than the number of simulation time instants) the first row of the workspace variable may be read again if the user has selected the cyclic read option (otherwise the output is 0).

The *MPSK Mod* block creates the complex envelope signal consisting of the real and imaginary parts of the signal. This block is a combination of the *MPSK Map* and *CE (Complex Envelope) PM* blocks. The digital input signals are in the range [0,M-1], where M is total number of different phases and the phase shift for input digit i is $2\pi i/M$. The MPSK Mod output is a unit-magnitude complex analog signal whose discrete values can be represented by M points on the unit circle in the complex plane.

The *MPSK Demod* block demodulates/demaps the complex input. This block is a combination of the *MPSK Corr Demod* and *CE Min/Max Demap* blocks. The MPSK Demod output is an integer in the range [0,M-1], where M is the number of phases.

The *AWGN (Additive White Gaussian Noise) Channel* block adds white Gaussian noise with user-specified mean and variance to the signal being transmitted through this channel. This block uses the *Gaussian White Noise Generator* block as the noise source. The vector length (N) for the initial seed entry determines the output vector size. The mean and the variance of the noise are specified such that the mean can be either a vector of length equal to the seed length (N) or a scalar, in which case all the elements of the noise vector share the same mean value. The covariance matrix can be one of the following:

- An N-by-N positive semi-definite matrix whose off-diagonal are the correlations
- A length N vector, in which case the individual elements of the noise vector are uncorrelated but have unequal variances

- A scalar, in which case all the elements of the noise vector are uncorrelated but share the same variance.

The *K-Step Delay* block delays its input (scalar) signal by K-sampling time intervals. This block function is equivalent to a z^{-k} discrete time operator.

The *Error Counter* block detects the difference between the port 1 signal and the port 2 signal at the times of the rising edge of the port 3 signal. If the difference between the port 1 and port 2 signals is larger than a user-defined tolerance, the error counter increments its output by 1. The rising edge of the signal at port 4 resets the error counter back to 0. The Error Counter block accepts scalar signals only.

The *Triggered Write to Workspace* block writes the input port 1 input variable to a workspace variable at the rising edge of the input port 2 trigger signal. There are six entries to this block: the workspace variable name, data type, the number of trigger pulses between saved data, the maximum row number, what to keep in the case of overflow, and the threshold value. The message signal to the input port 1 can be either a scalar or a vector while the trigger signal to the input port 2 must be a scalar. The saved workspace variable is a column vector if port 1 input is a scalar and a matrix if the port 1 input is a vector. In the later case the first element of the input signal vector is saved in the first column, the second element of the input signal vector is saved in the second column, and so on. The block saves the first record at the first rising edge of the trigger signal. The data can be saved either as strings or data. A pre-defined number limits the maximum row of the output variable (this number cannot be changed during simulation).

IV. BIT ERROR PROBABILITY CONVERGENCE

In this chapter the convergence of the bit error probability for BPSK with AWGN is investigated. The probability of bit error obtained by simulation converges to the theoretical values as the "transmitted" number of symbols (and the simulation run time) increases. Convergence was tested for signal-to-noise (SNR) ratios with noise variances (the signal power was kept constant) of 0.5, 1, 2, 4, 9, and 16. The signal amplitude is denoted A (A=1), the symbol duration is denoted T (T=1s), the sampling interval is denoted Δt (Δt=T/4=0.25s), and the noise variance is denoted σ^2. With this notation, the signal-to-noise ratio SNR in dB and the theoretical values of the bit error probability for BPSK are:

$$SNR_{(dB)} = 10\log(\frac{A^2T}{2\sigma^2\Delta t}) \tag{23}$$

$$P_B = Q(\sqrt{10^{2\frac{SNR_{dB}}{10}}}) \tag{24}$$

The theoretical values, calculated using Equations 23 and 24, are shown in Table II. Before showing the convergence results for each variance/SNR case, it is useful to explain the results for one variance. For example, Table I shows the convergence results for the noise variance of unity. The table has the columns for:

- the number of symbols used in the simulation
- the theoretical value of the bit error probability
- the bit error probability obtained by the simulation (for the corresponding number of symbols "transmitted") , and

- the difference between the theoretical and the simulation bit error probabilities.

The table shows that the simulation result converges to the theoretical result as the number of "transmitted" symbols increases.

Number of Symbols	**Theory**	**Simulation**	**Difference in BER**
1000	0.0230	0.0250	8.69%
10000	0.0230	0.0237	3.04%
100000	0.0230	0.0233	1.30%
1000000	0.0230	0.0231	0.43%

Table I. Simulation Results for $\sigma^2 = 1$

The estimated bit error probability as a function of the symbol number for each of the simulations (from 10^3 to 10^6 symbols) is shown in Figures 4 through 7.

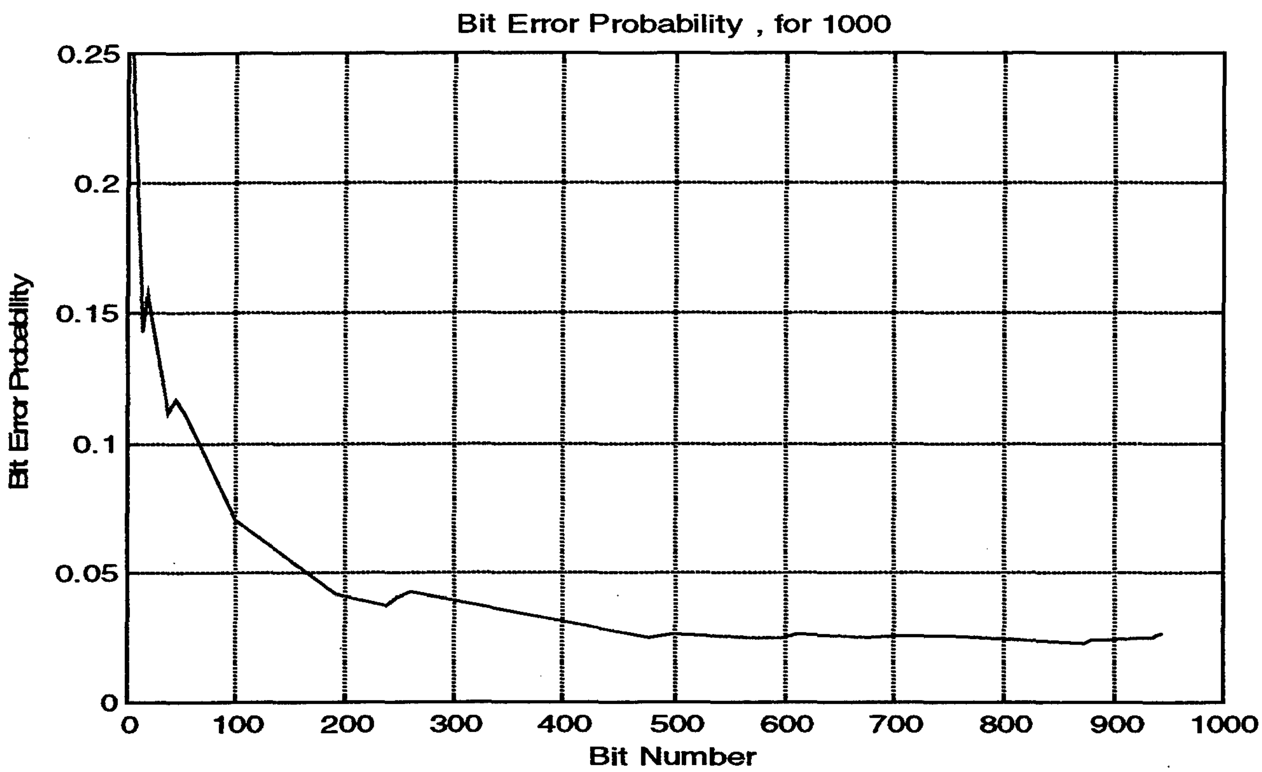


Figure 4. BER Convergence for 10^3 symbols

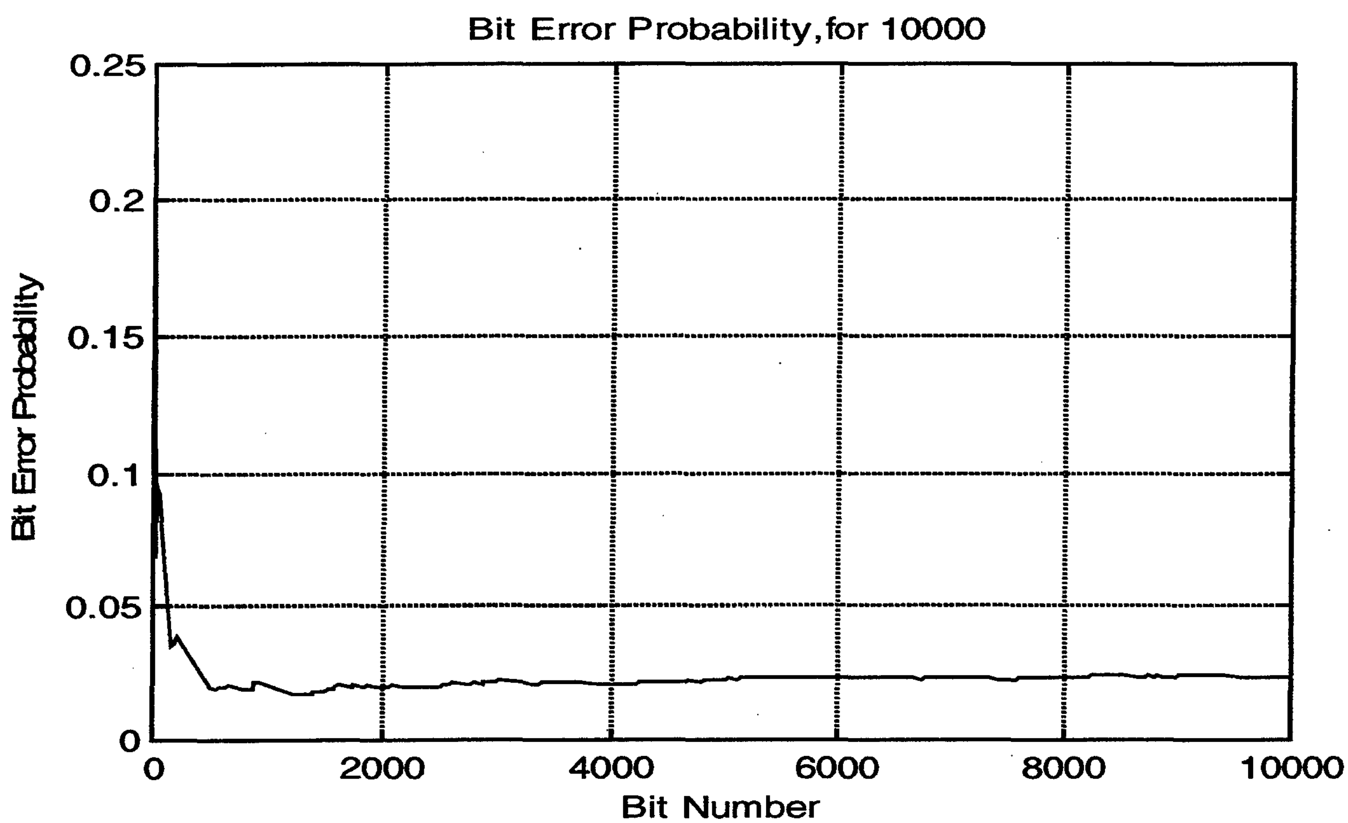


Figure 5. BER Convergence for 10^4 symbols

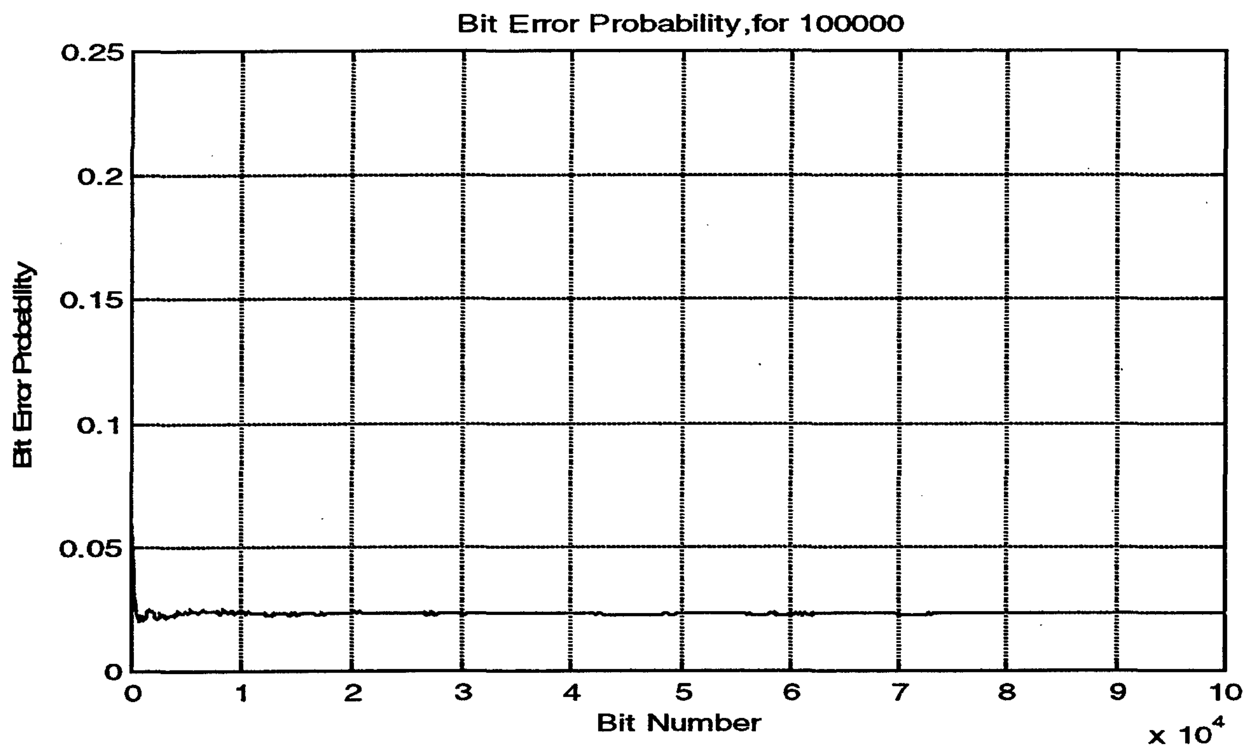


Figure 6. BER Convergence for 10^5 symbols

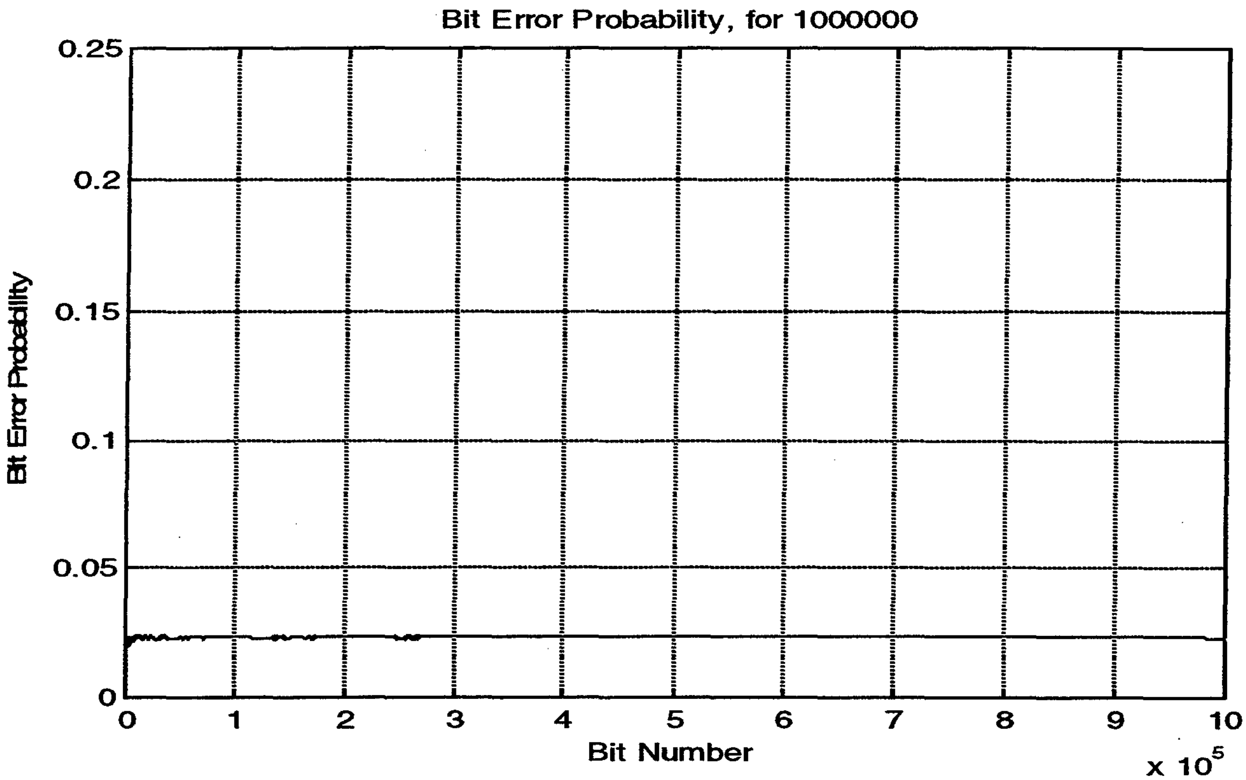


Figure 7. BER convergence for 10^6 symbols

For the case of 1,000 symbols, the simulation gives the estimate of the probability of error as 0.025, as seen in Figure 4. Note that each time a simulation is repeated for the same number of symbols and the same variance but with a different seed value for the noise generator, a different result is in principle obtained. However, the average of all such simulation results becomes essentially a constant as the number of repeated simulations increases. When the number of transmitted symbols is increased to 10,000, a different estimate for the bit error probability is obtained, 0.0237. If the number of transmitted symbols is increased to 100,000, the bit error probability obtained is 0.0233 and there is very little change past 3,000 or so symbols. Lastly, increasing the number of transmitted symbols to 1,000,000 yields the most accurate estimate for the bit error probability of 0.0231.

Next, the number of transmitted symbols was selected as 10^6 and the simulation was run for the noise variances between 0.5 and 16. The results for the estimates of the bit error probability and their difference relative to the theoretical bit error probabilities are shown in Table II.

Noise Variance	Theoretical P_b Value	Simulation P_b Value	Number of Trials	Difference
0.5	$2.339*10^{-3}$	$2.4*10^{-3}$	1000000	2.61%
1.0	0.0230	0.0231	1000000	0.43%
2.0	0.0790	0.0790	1000000	0%
4.0	0.1590	0.1590	1000000	0%
9.0	0.2520	0.2529	1000000	0.36%
16.0	0.3090	0.3087	1000000	0.097%

Table II. Simulation Results for Various Variances

The convergence of the bit error estimates as functions of the numbers of transmitted symbols is shown in Figures 8 through 12.

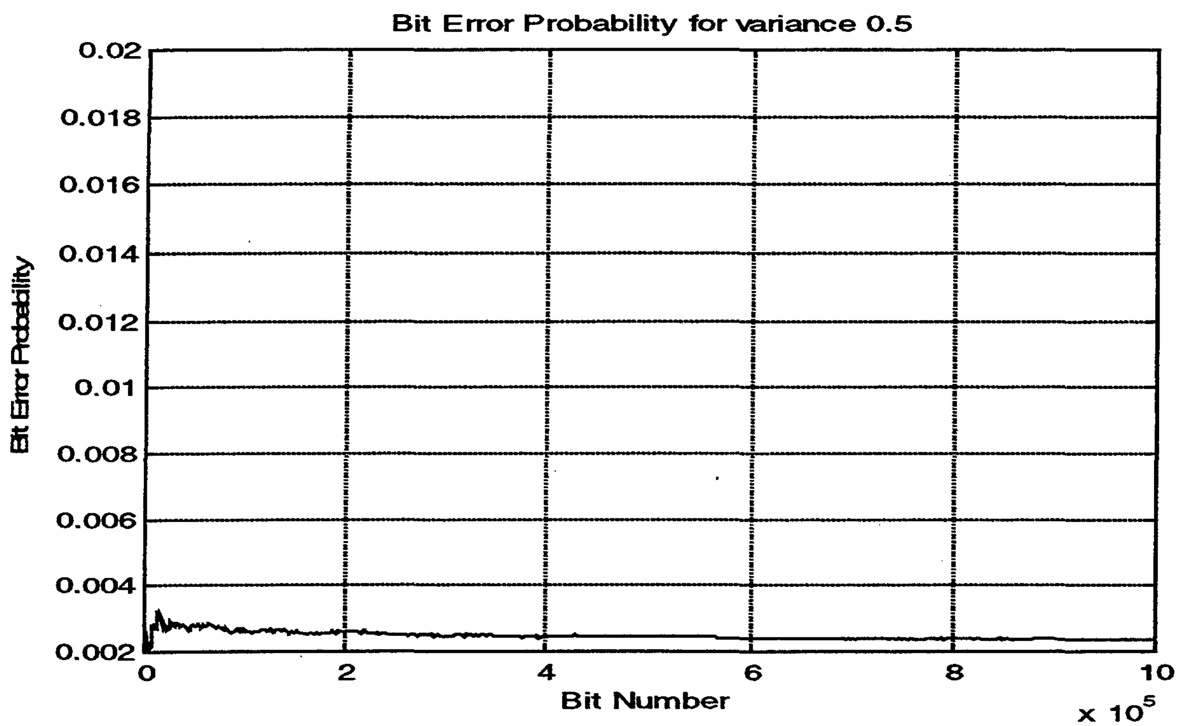


Figure 8. BER Convergence for $\sigma^2 = 0.5$

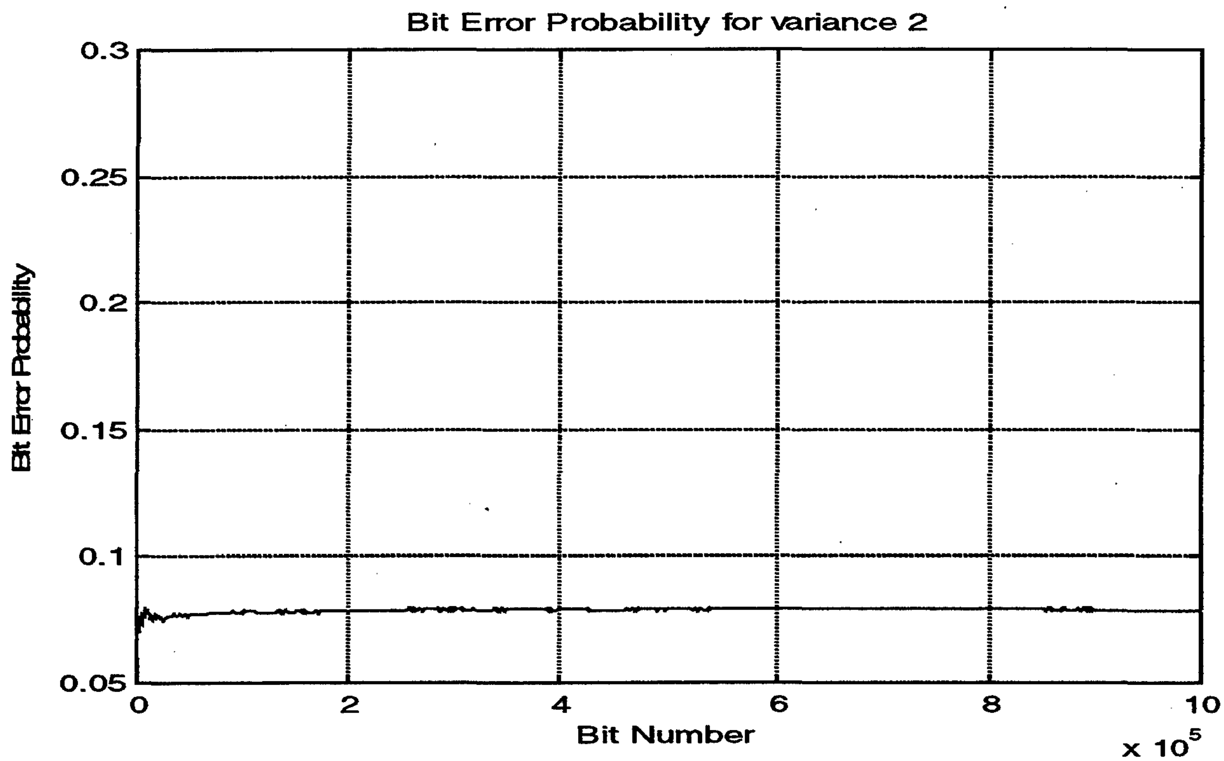


Figure 9. BER Convergence for $\sigma^2 = 2$

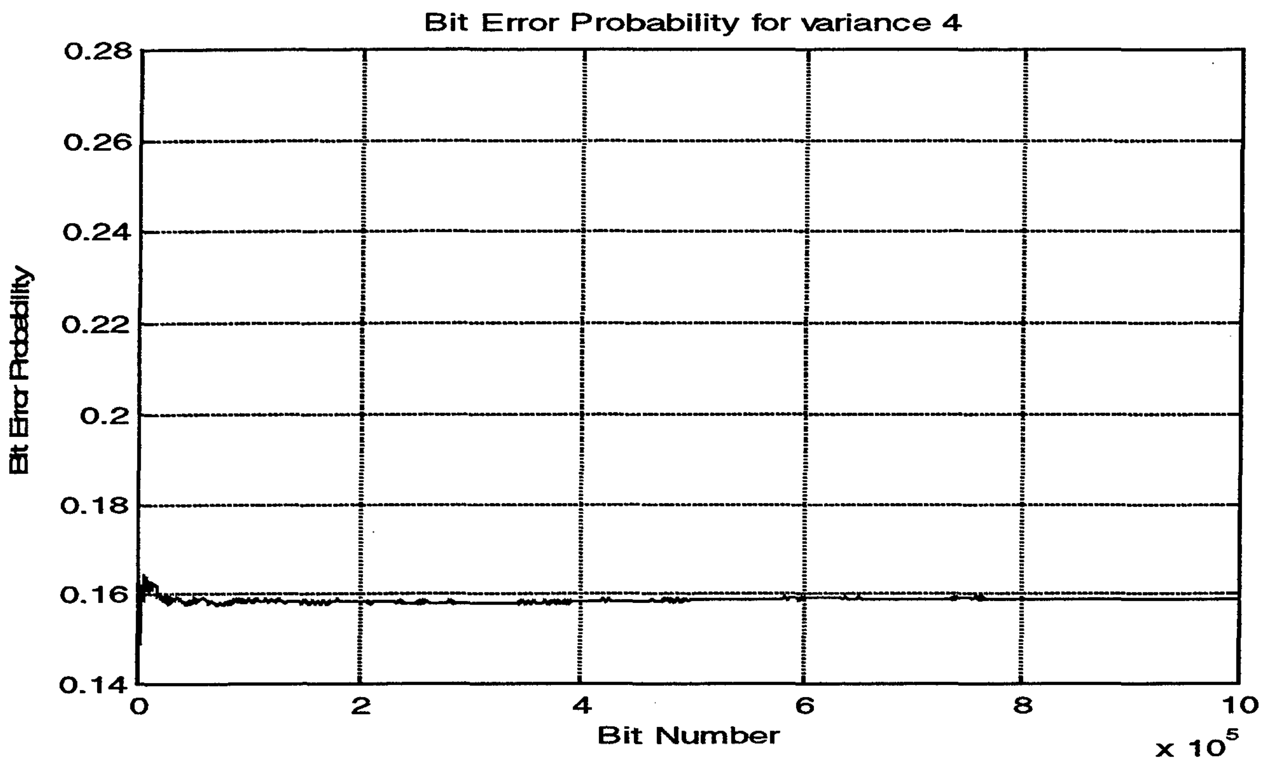


Figure 10. BER Convergence for $\sigma^2 = 4$

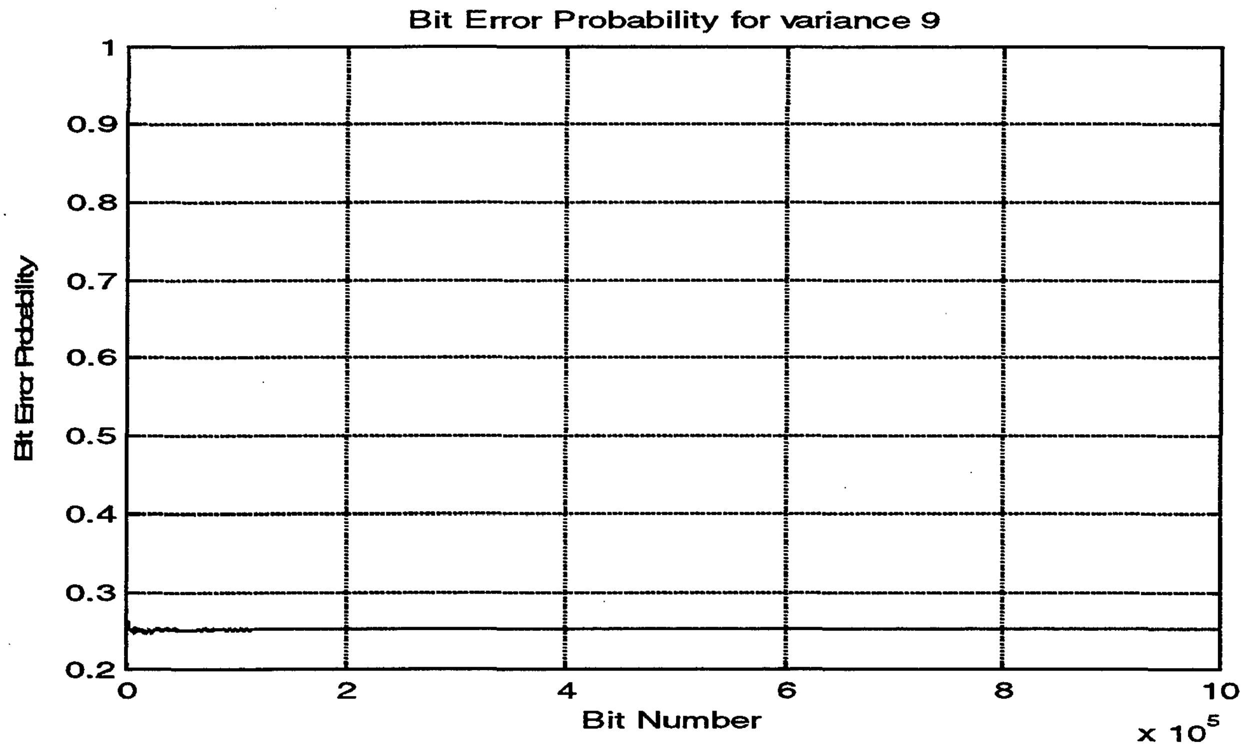


Figure 11. BER Convergence for $\sigma^2 = 9$

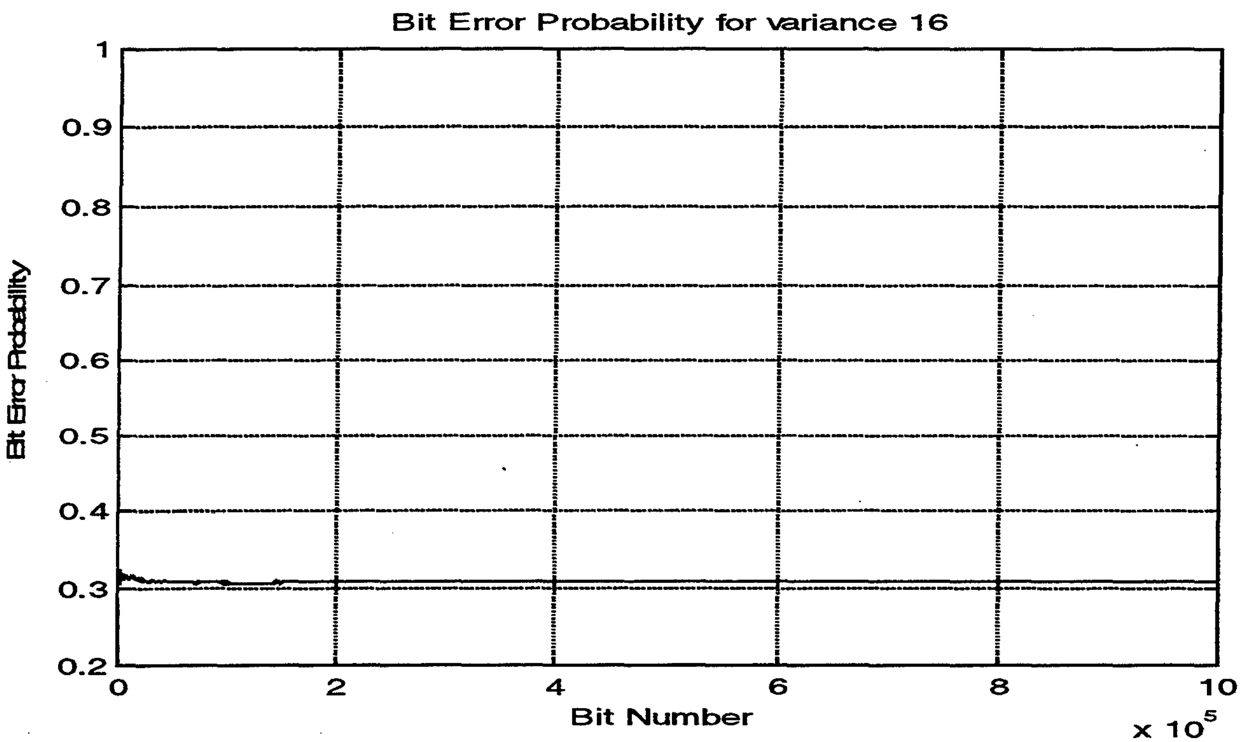


Figure 12. BER Convergence for $\sigma^2 = 16$

The bit error probability estimates obtained by simulation for various SNR's are shown in Figure 13 against the theoretical bit error probability for BPSK with AWGN.

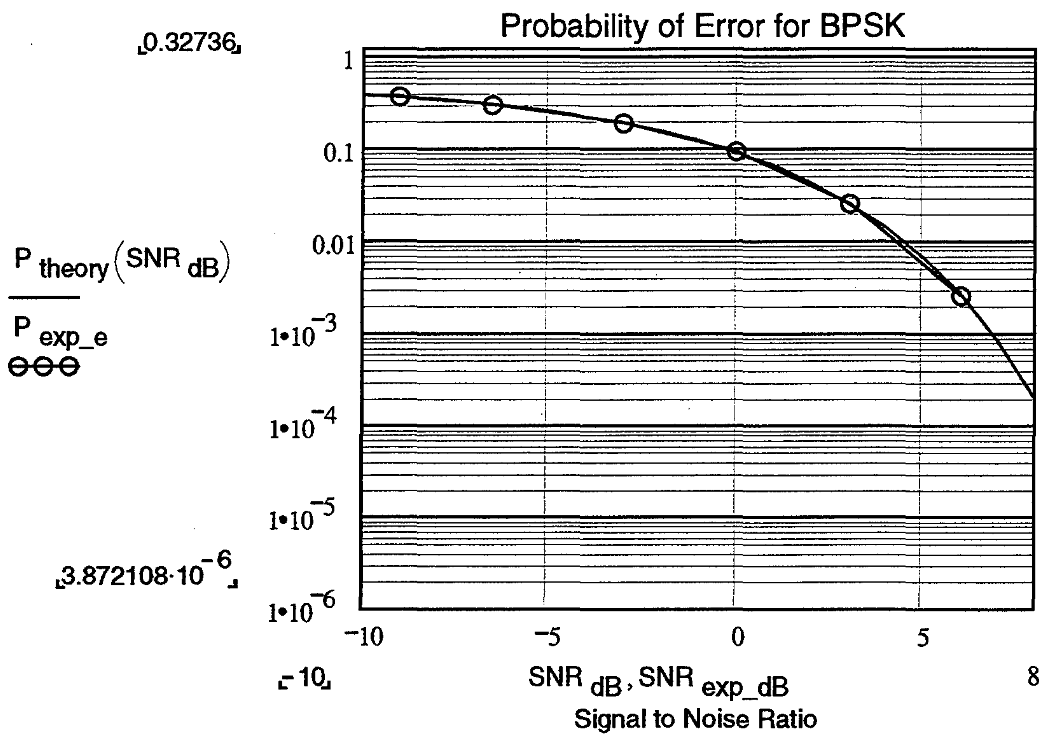


Figure 13. BER vs. SNR

As can be seen, the theoretical bit error probability and the Simulink results match each other closely.

V. SIMULATION VERIFICATION FOR BPSK AND QPSK WITH AWGN

In this chapter the simulation results obtained from the (modified) Simulink model and a Matlab program will be presented and discussed. The Simulink model was modified to include co-channel interference (discussed in the next chapter). In this chapter, only the AWGN will be considered (the co-channel interference power was set to 0) for BPSK and QPSK modulations. The estimates for the bit error probability were obtained using both the Simulink model shown in Figure 14 and the Matlab program mpsk1.m listed in the Appendix M. The mpsk1.m program uses Communications Toolbox functions rather than Simulink blocks to implement the simulations. Although a time-domain simulation of a communication system is in general possible using either Simulink block-diagrams or the equivalent Matlab functions, there is a very important difference between the two. The essential feature of Simulink is that simulations progress one step at the time, that is the input data is processed sequentially (one at a time), and are referred to as "point data" or "time-flow" simulations. On the other hand, Matlab simulations can process vectors ("blocks") of data at each simulation step and are thus referred to as the "block-data" or "data-flow" simulations. Block-data simulations are substantially faster than point-data simulations. However, block-data simulations cannot be applied in the time-domain to simulate feedback-systems since such systems require the responses to be calculated at each time instant before new inputs are applied.

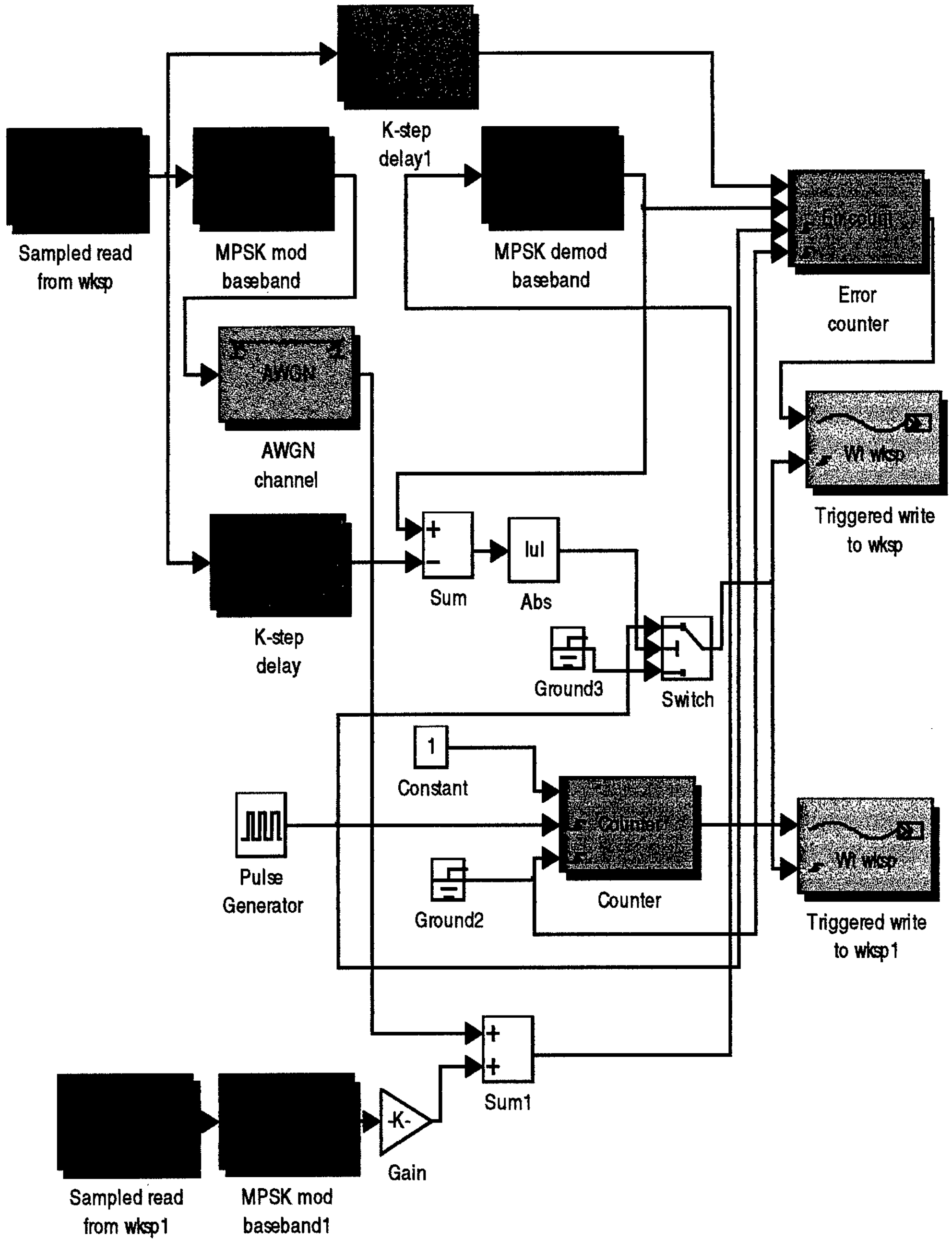


Figure 14. Simulink MPSK Model with Co-channel Interference

For both BPSK and QPSK, only AWGN was considered and the co-channel interference was made negligible by setting the signal-to-interference ratio to +100 dB. Bit signal-to-noise (SNR) ratios in the range from –5 dB to +12 dB were used in the simulations.

The simulation results were plotted using auxiliary Mathcad "scripts":

- plot_sim_noise2.mcd for Simulink BPSK simulations
- plot_mat_noise2.mcd for Matlab BPSK simulations
- plot_sim_noise4.mcd for Simulink QPSK simulations and
- plot_mat_noise4.mcd for Matlab QPSK simulations.

A. BPSK BIT ERROR PROBABILITY ESTIMATES

1. BPSK Results for the Simulink Model

The Simulink model was run from a Matlab script in order to automate the data input and data output operations for simulations run for multiple values of signal-to-noise ratio SNR. The input data file is created using Matlab program prep.m. Another Matlab program, sun_psk.m, reads the input data file, starts the simulation (executed in a loop for various values of SNR), and saves the simulation output data. The input data used for the simulation (and defined by running the prep.m Matlab script) are:

Noise (1) or Noise and Interference (2): 1

Symbol Duration: 1s

Oversampling Factor: 2

Minimum Bit Signal to Noise Ratio: -5 dB

Maximum Bit Signal to Noise Ratio: 12 dB

Number of values for SNR: 10

Minimum number of errors acceptable: 100

Factor multiplying the error numbers: 2

Maximum size of the random integer arrays: 10^6

File name to save data: SIMULINK_NOISE_2

The output data saved in the ASCII format includes:

- bit error probabilities (BER)
- signal-to-noise ratios (SNR)
- signal-to-interference ratios (SJR) (set to +100 dB for the noise-only case)
- numbers of errors observed
- numbers of transmitted symbols.

The estimates (obtained from the Simulink model) for the BPSK bit error probability are shown in Figure 15 together with the theoretical BER versus the SNR .

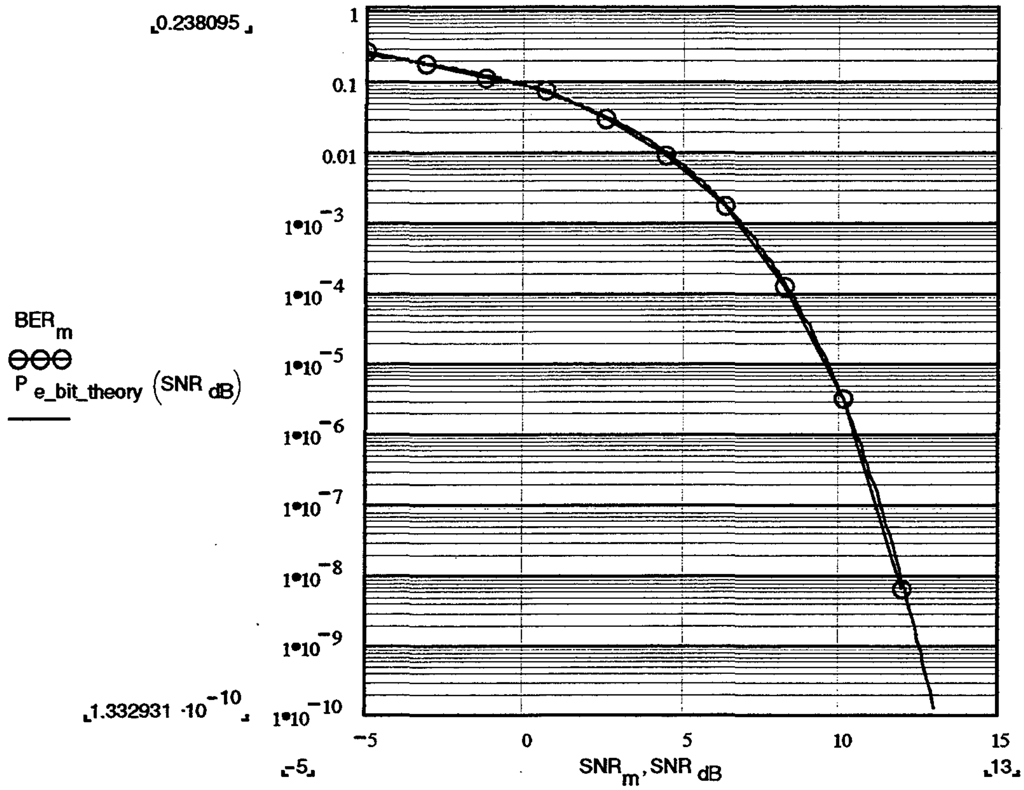


Figure 15. BPSK BER versus SNR (Theory and Simulink Model Estimates)

2. BPSK Results Obtained by Using Matlab Program MPSK1.M

The same MPSK communication system that was modeled in Simulink using Simulink and Communications Toolbox blocks was implemented in Matlab using Matlab and Communications Toolbox functions. The same parameters were used for the simulation:

Noise (1) or Noise and Interference (2): 1

Symbol Duration: 1s

Oversampling Factor: 2

Minimum Bit Signal to Noise Ratio: -5 dB

Maximum Bit Signal to Noise Ratio: 12 dB

Number of values for SNR: 10 or 18

Minimum number of errors acceptable: 100

Factor multiplying the error numbers: 2

Maximum size of the random integer arrays: 10^6

File name to save data: MATLAB_NOISE_2

The simulation output data were saved and visualized using an auxiliary Mathcad script. The theoretical bit error probability and the estimates obtained from the Matlab simulation are shown in Figure 16 as functions of the signal-to-noise ratio SNR.

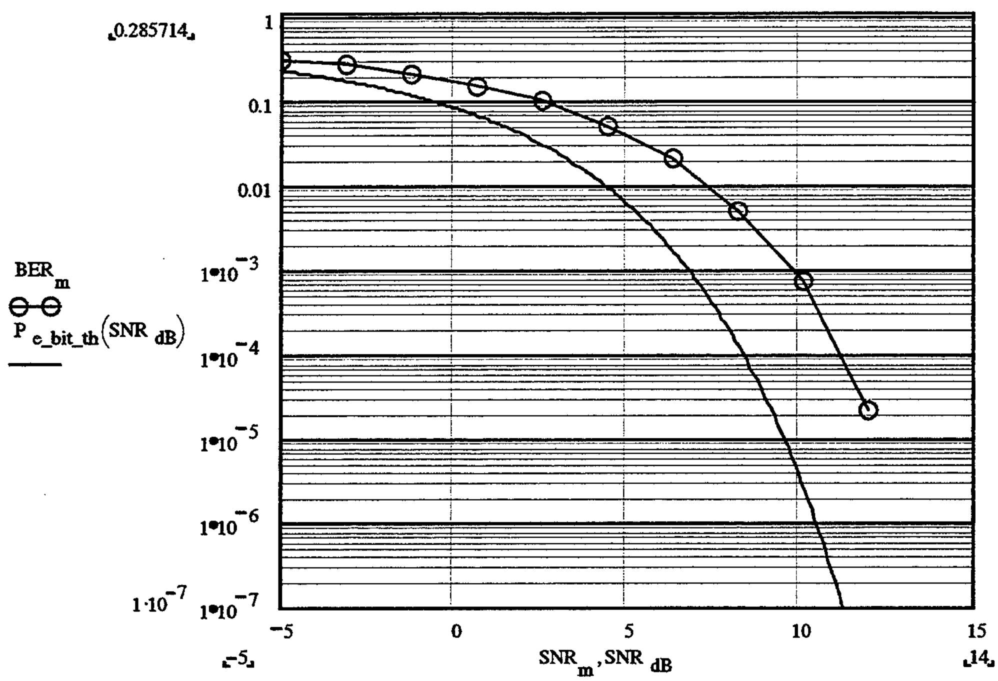


Figure 16. BPSK BER versus SNR (Theory and Matlab Model Estimates)

In this case, simulation results differ from theory. At closer inspection, one notices that the curve for the estimates is similar to the theoretical curve but shifted to the right. Such a shift corresponds to an SNR "loss". Indeed, if the SNR for the theoretical curve is reduced by 3 dB, the theoretical curve and the Matlab estimate match each other, as shown in Figure 17. Therefore, we may conclude that the Simulink model gives very accurate estimates for the bit error rate of BPSK with AWGN while the Matlab model experiences a 3 dB SNR loss (compensating for this SNR loss again produces accurate estimates). This seems to indicate a fundamental problem with the Communications Toolbox.

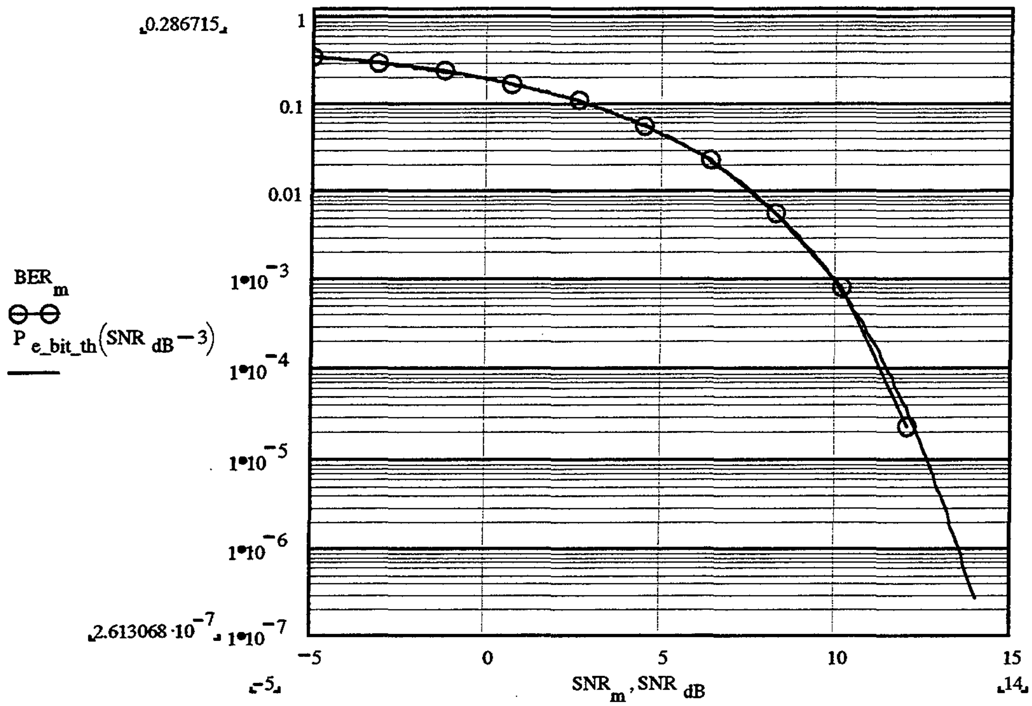


Figure 17. BPSK BER versus SNR (Theory for SNR-3dB and Matlab for SNR)

B. QPSK BIT ERROR PROBABILITY ESTIMATES

1. QPSK Results for the Simulink Model

The Simulink model used for the BPSK is a general model that can be used for any MPSK. Therefore, the same model was used for QPSK by selecting the number of phases M (an input parameter) as four. The Simulink model results for the QPSK bit error probability estimates are shown in Figure 18, together with the theoretical bit error probability and its union bound given by [Ref.4]

$$P_{e\_bit\_ub}(M, SNR_{bit\_dB}) = \frac{M}{M-1} Q\left(\sqrt{\frac{\log(M)}{\log(2)} 2.10^{\frac{SNR_{bit\_dB}}{10}}} . \sin\left(\frac{\pi}{M}\right)\right) \tag{25}$$

as functions of the *bit* signal-to-noise ratio SNR_{bit}. We note that the estimates differ somewhat from theory but that this difference diminishes as the bit SNR increases.

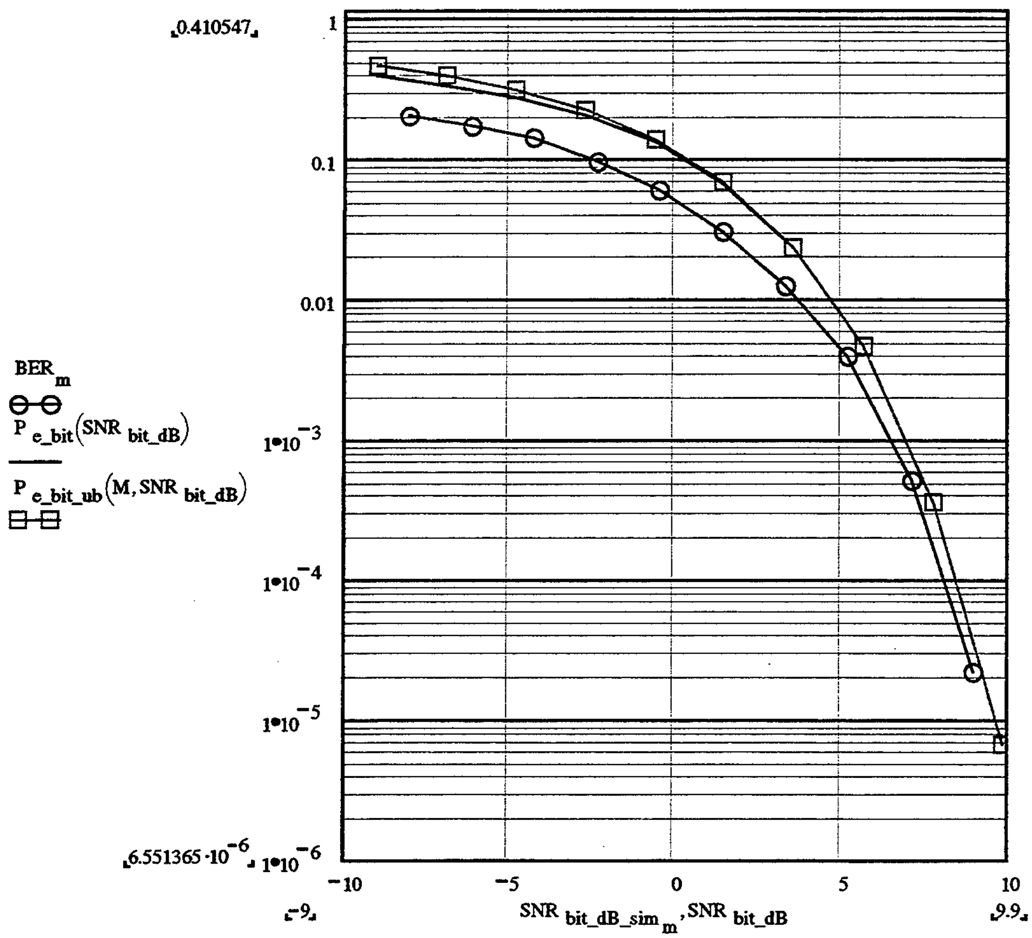


Figure 18. QPSK BER versus SNR (Theory and Simulink Model Estimates)

2. QPSK Results Obtained by Using Matlab Program MPSK1.M

The estimates for QPSK BER obtained using Matlab are shown in Figure 19, together with the theoretical bit error probability and its union-bound estimate given by Equation 25 as functions of the bit SNR.

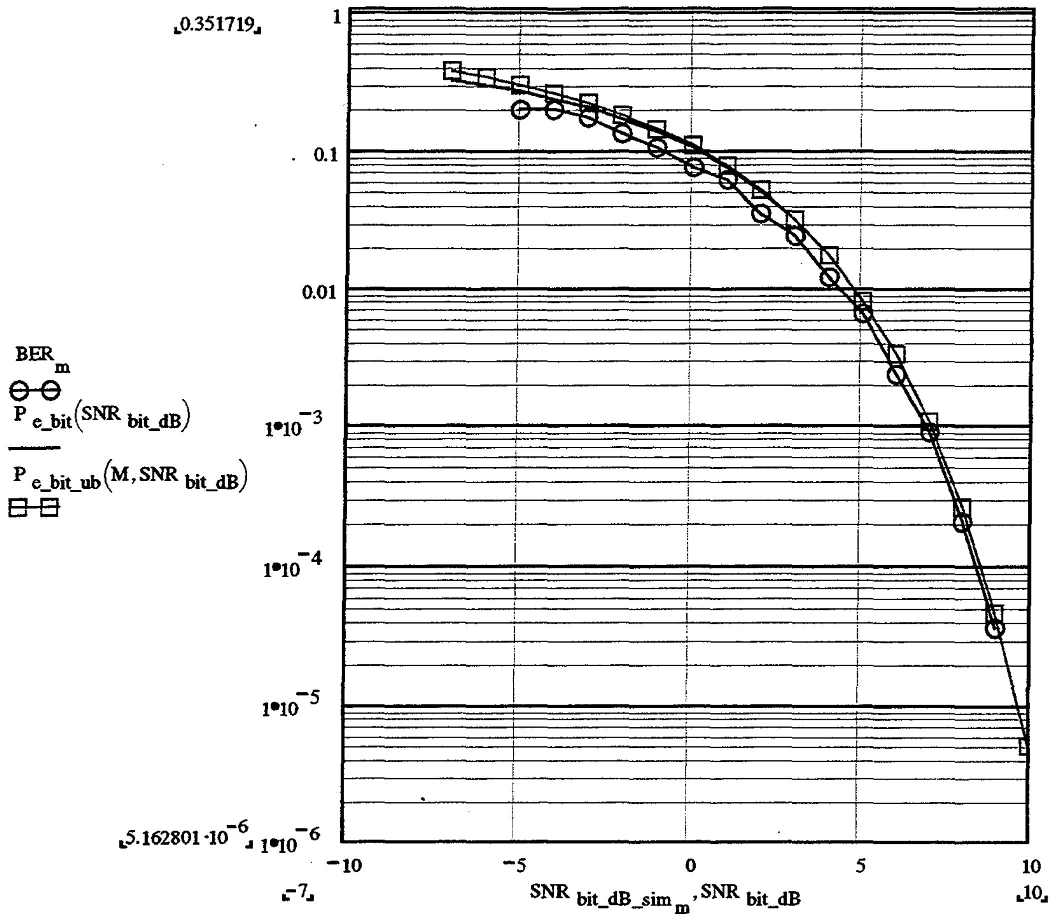


Figure 19. QPSK BER versus SNR (Theory and Matlab Model Estimates)

VI. SIMULATION VERIFICATION FOR BPSK AND QPSK WITH ADDITIVE WHITE GAUSSIAN NOISE AND CO-CHANNEL INTERFERENCE

A. BPSK BIT ERROR PROBABILITY ESTIMATES

In the preceding chapter only additive white Gaussian noise (AWGN) was considered for BPSK and QPSK simulations. In this chapter both AWGN and co-channel interference are considered. A second MPSK Mod block fed by an independent message source generates the co-channel interference and is added to the sum of the signal and noise. In the Simulink model and the Matlab program a range of signal-to-noise ratios (SNR) and signal-to-interference ratios (SJR) are used to calculate the estimates of the bit error probability as functions of SNR and SJR for both BPSK and QPSK. The results are presented as families of curves, one with SNR as the variable and SJR as the parameter and the other with SJR as the variable as SNR as the parameter.

The parameters for the BPSK simulations using both the Simulink model and Matlab were:

Noise (1) or Noise and Interference (2): 2

The Number of Phases: 2

Symbol Duration: 1s

Oversampling Factor: 2

Minimum Bit Signal to Noise Ratio: -5 dB

Maximum Bit Signal to Noise Ratio: 12 dB

Minimum Bit Signal to Interference Ratio: -5 dB

Maximum Bit Signal to Interference Ratio: 12 dB

Number of values for SNR: 10

Number of values for SJR: 10

Minimum number of errors acceptable: 100

Factor multiplying the number of error: 2

Maximum size of the random integer arrays: 10^6

File name to save data: NOISE&INTERFERENCE_2

The probabilities of bit error were saved as a BER matrix whose rows correspond to constant values of SJR and whose columns correspond to constant values of SNR. The selected ranges for SNR and SJR were from –5 dB to +12 dB. For each bit error probability estimate (each matrix entry) at least 100 bit errors were observed. The simulations ran until the error counter exceeded the minimum specified number of error, in this case 100. Upon exceeding the minimum number of required bit errors, the simulation was restarted for another combination of SNR and SJR (to create another BER matrix entry). High values of SNR or SJR correspond to low noise and interference powers, which in turn implies low probabilities of bit error. Therefore, the time required for the simulation to produce at least 100 errors increases as the row and column indices (SJR and SNR) increase, and the BER matrix entries in general decrease with increasing the row and column indices.

For QPSK simulations the same parameter values were used as for BPSK except for the number of phases M, which are four for QPSK. The BPSK simulation results were plotted using auxiliary Mathcad programs plot_sim_noise&int2.mcd for the Simulink model and plot_mat_noise&int2.mcd for Matlab program results. Similarly, the QPSK simulation results were plotted using auxiliary Mathcad programs plot_sim_noise&int4.mcd for Simulink model and plot_mat_noise&int4.mcd for the Matlab program.

1. BPSK Results for the Simulink Model

The results for the bit error probability of BPSK with noise and co-channel interference obtained using the Simulink model are shown in Figure 20 (BER versus SNR with SJR as parameter) and in Figure 21 (BER versus SJR with SNR as parameter). The increments for both SNR and SJR are 1.889 dB, starting from –5 dB.

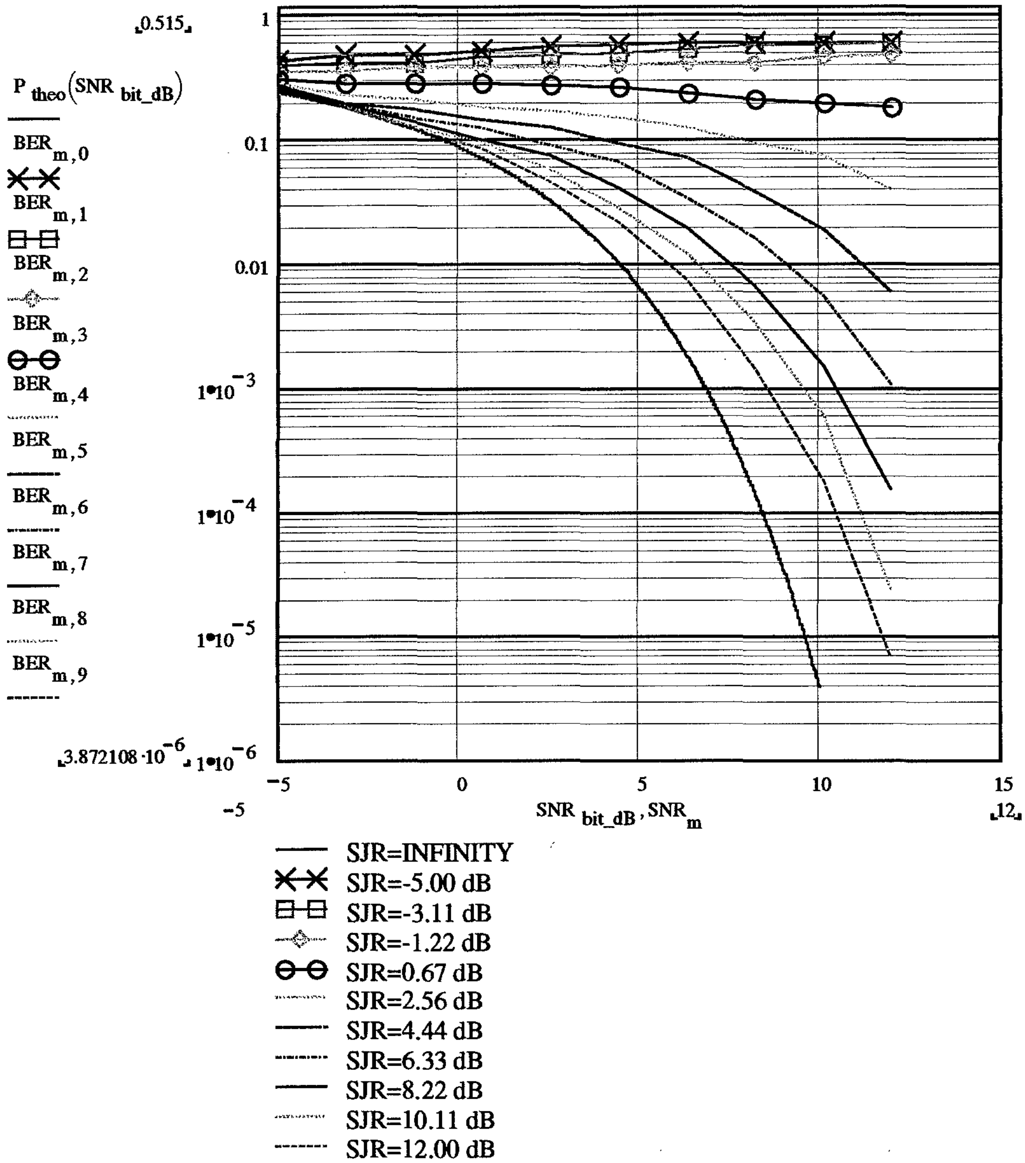


Figure 20. BPSK BER versus SNR with SJR as Parameter (Simulink Model Estimates)

The "lowest" curve represents the theoretical bit error probability for BPSK with AWGN only. The increase of BER with decreasing SJR is evident from Figure 20. The BPSK probability of bit error as a function of SJR, with SNR as a parameter starting at –5 dB and with 1.889-dB increments, is shown in Figure 21.

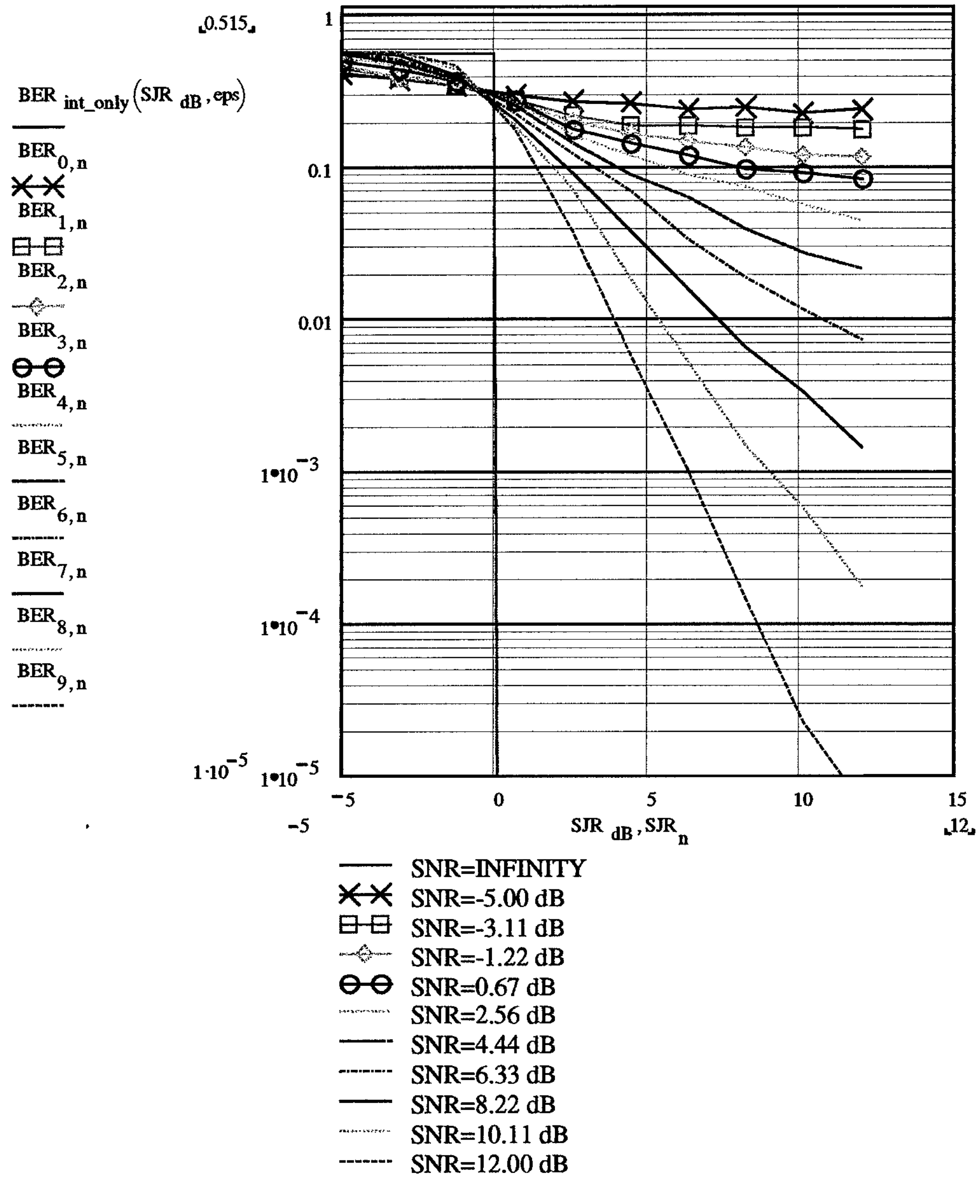


Figure 21. BPSK BER versus SJR with SNR as Parameter (Simulink Model Estimates)

The "step function" represents the noise-free case, that is the case of co-channel interference only. In such a case the probability of error is either 0.5 when the co-channel interference power is larger than the signal power (negative SJR) or 0 when the signal power is larger than the co-channel interference power. (In order to use the logarithmic scale, the value of 0 for positive SJR has been replaced by a small non-zero value).

2. BPSK Results Obtained by Using Matlab Program MPSK1.M

The results for the bit error probability of BPSK with noise and co-channel interference obtained using Matlab program are shown in Figures 22 and 23 (BER versus SNR with SJR as parameter starting at –5 dB and with 1.889 dB increments) and in Figure 24 (BER versus SJR with SNR as parameter starting at –5 dB and with 1.889 dB increments). Almost 3 dB difference can be noticed between simulation results and the theoretical value. This difference was explained in the preceding section.

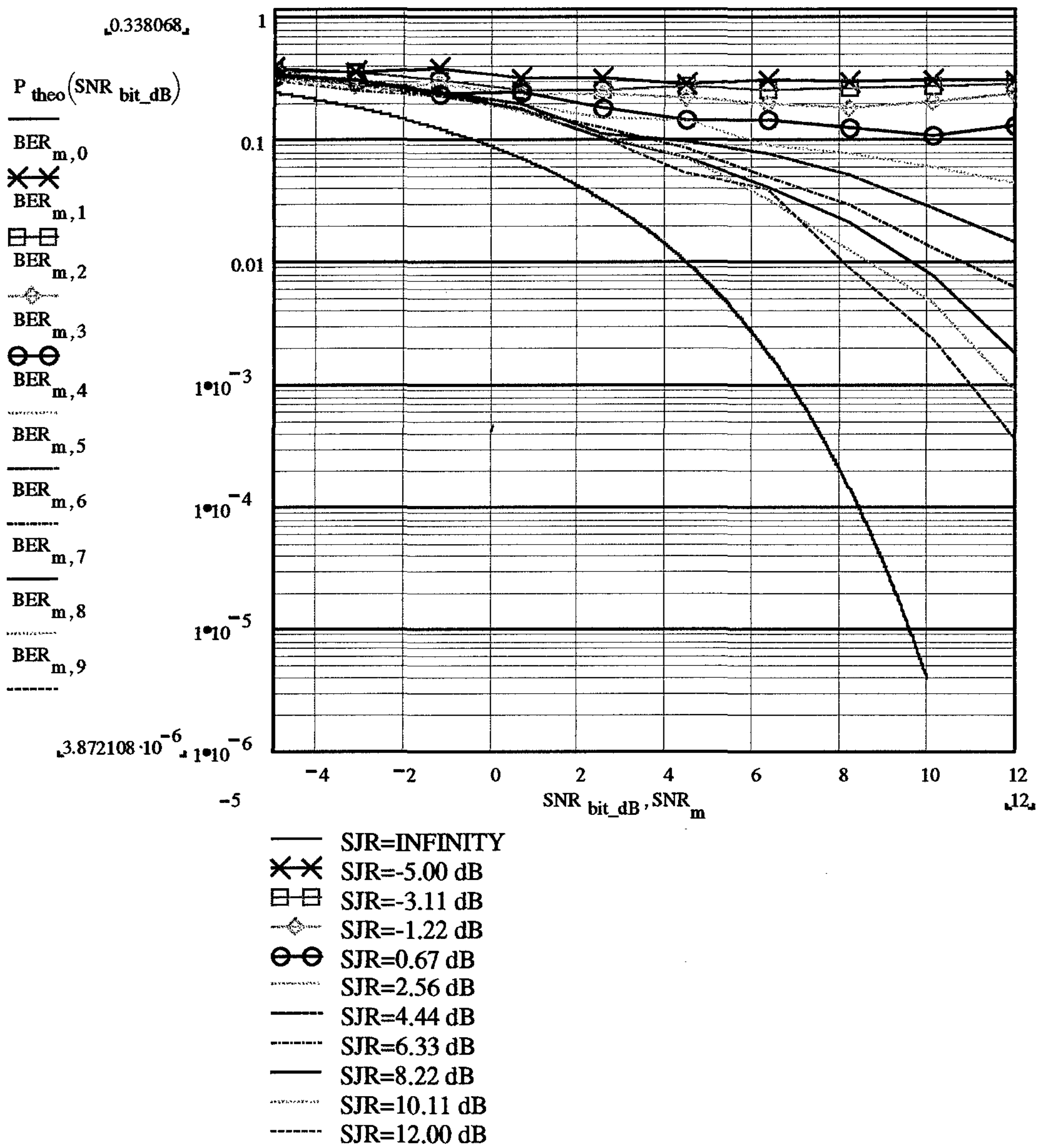


Figure 22. BPSK BER versus SNR with SJR as Parameter (Matlab Model Estimates)

Figure 23 shows the BPSK probability of bit error as a function of SJR, with SNR as a parameter. Again, the difference between SNR curves is 1.889 dB as in SJR case. The "step function" again represents the noise-free case, that is the case of co-channel interference only.

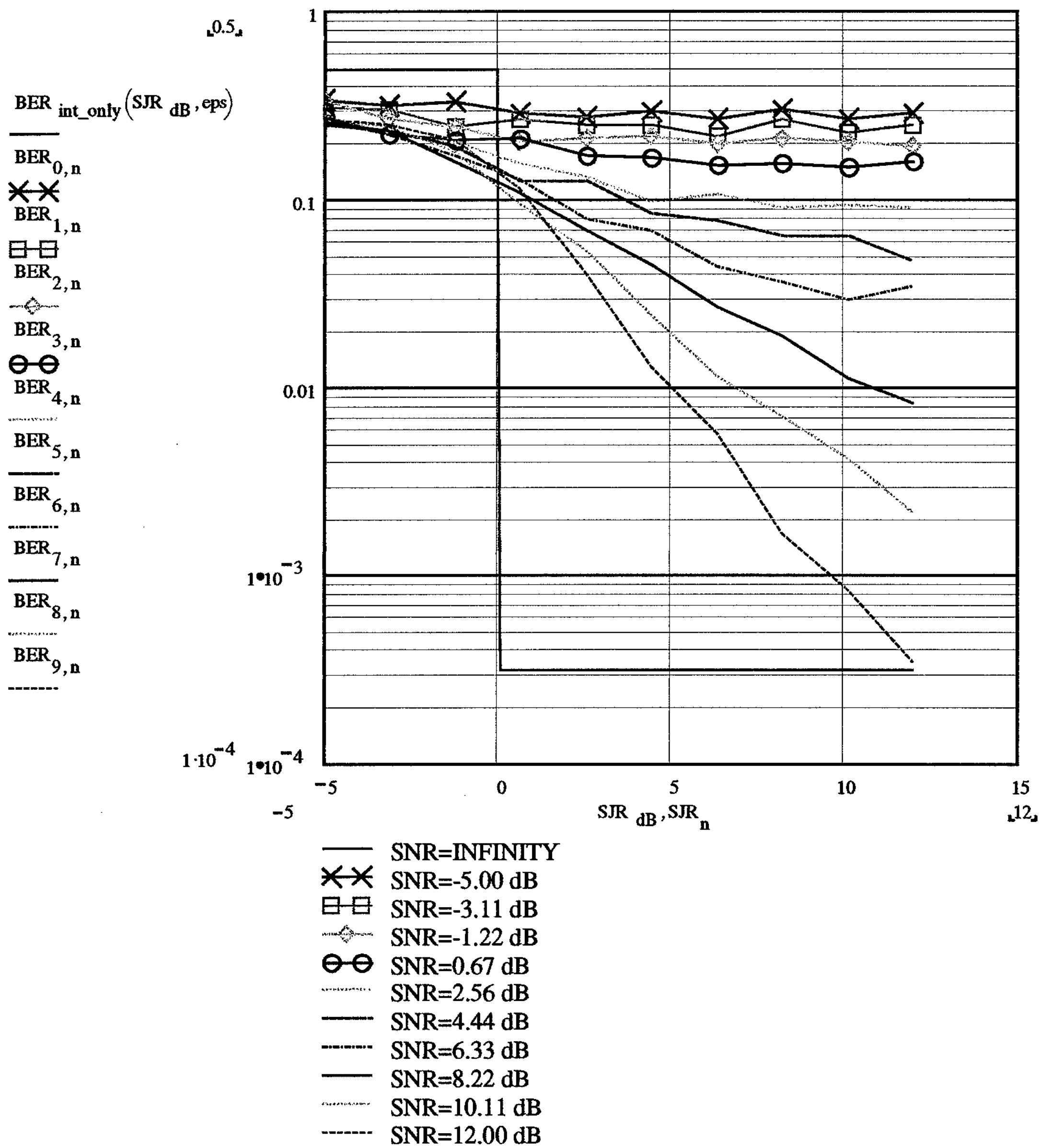


Figure 23. BPSK BER versus SJR with SNR as Parameter (Matlab Model Estimates)

B. QPSK BIT ERROR PROBABILITY ESTIMATES

1. QPSK Results for the Simulink Model

The results for the bit error probability of QPSK with noise and co-channel interference obtained using the Simulink model are shown in Figure 24 (BER versus SNR with SJR as parameter starting at –5 dB and with 1.889 dB increments) and in Figure 25 (BER versus SJR with SNR as parameter starting at –5 dB and with 1.889-dB increments).

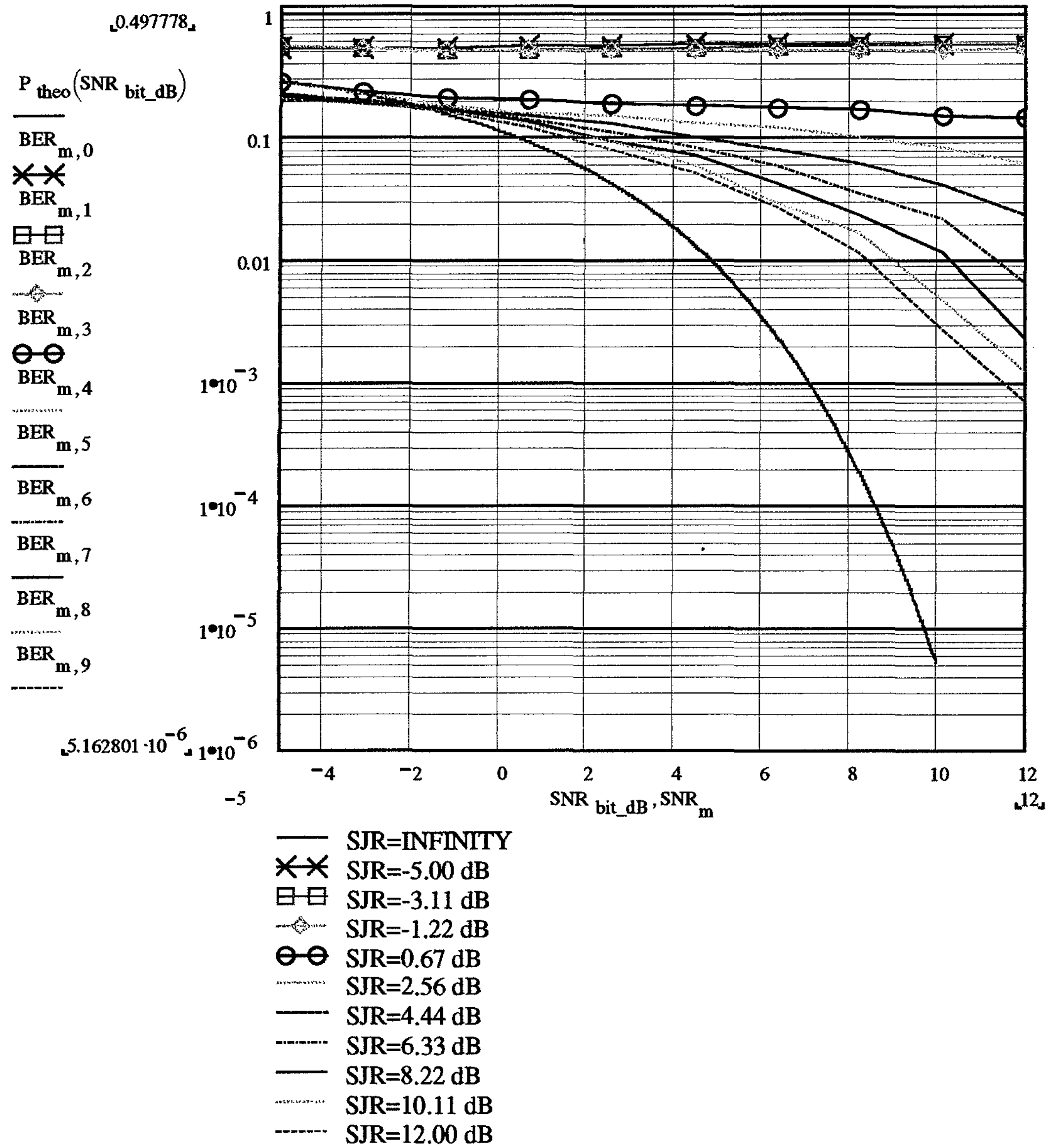


Figure 24. QPSK BER versus SNR with SJR as Parameter (Simulink Model Estimates)

The "lowest" curve represents the theoretical bit error probability for QPSK with AWGN only. The increase of BER with decreasing SJR (for positive SNR) is evident from Figure 24.

The QPSK probability of bit error as a function of SJR, with SNR as a parameter starting at –5 dB and with 1.889-dB increments, is shown in Figure 25. The "step function" again represents the noise-free case, that is the case of co-channel interference only.

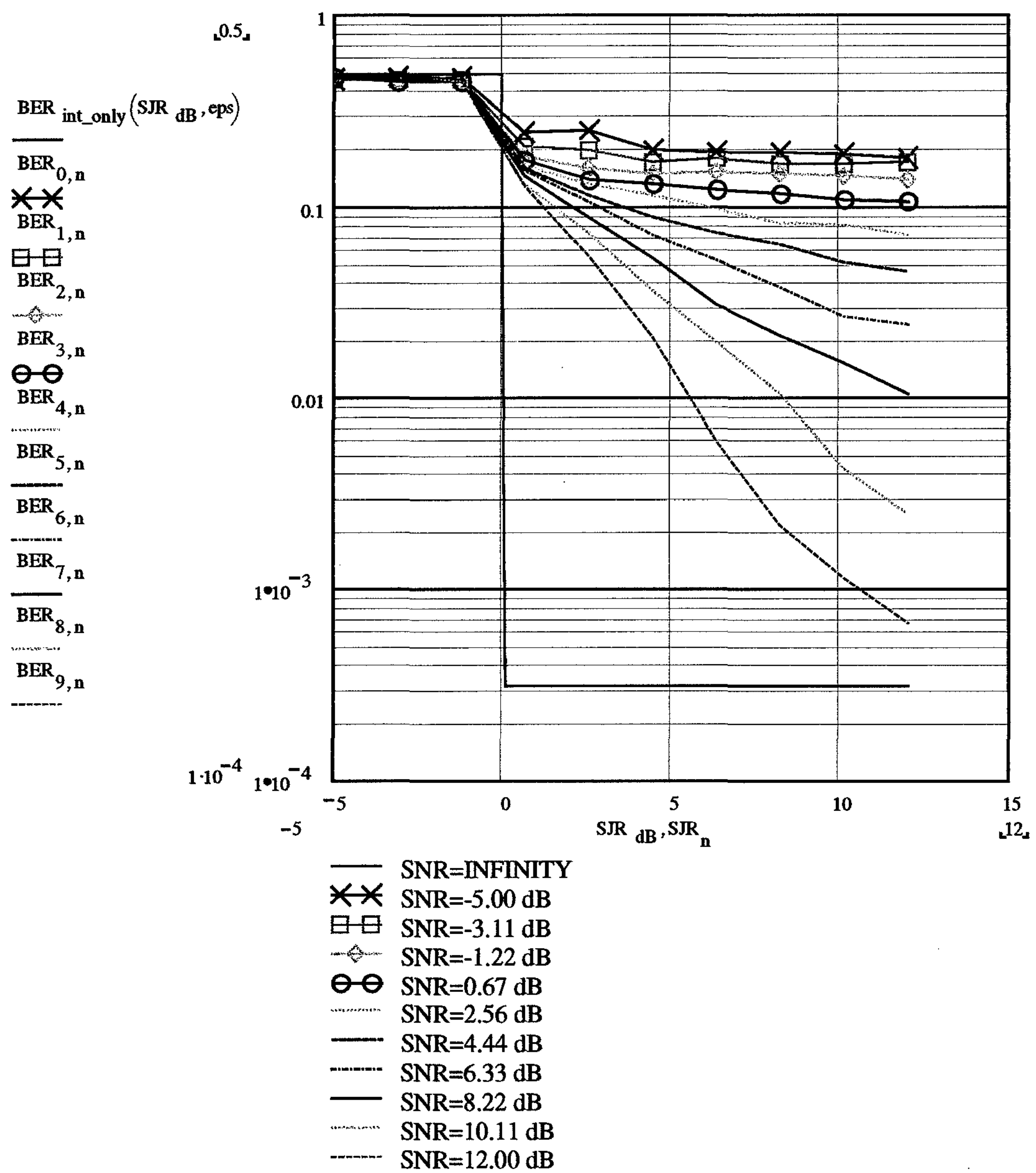


Figure 25. QPSK BER versus SJR with SNR as Parameter (Simulink Model Estimates)

As can be seen in Figure 25, the curves for the bit error probability tend to the noise-free step function as the SNR increases. Also, the curves for the negative SJR tend to 0.5, meaning that the bit error probability is dominated by the interference for negative SJR, as expected.

2. QPSK Results Obtained by Using Matlab Program MPSK1.M

The estimates for QPSK BER obtained using Matlab are shown in Figure 26 together with the theoretical bit error probability as a function of the bit SNR and with SJR as parameter starting at –5 dB and with 1.889 dB increments. Comparing the BER curves in Figures 24 and 26, we note that they differ slightly.

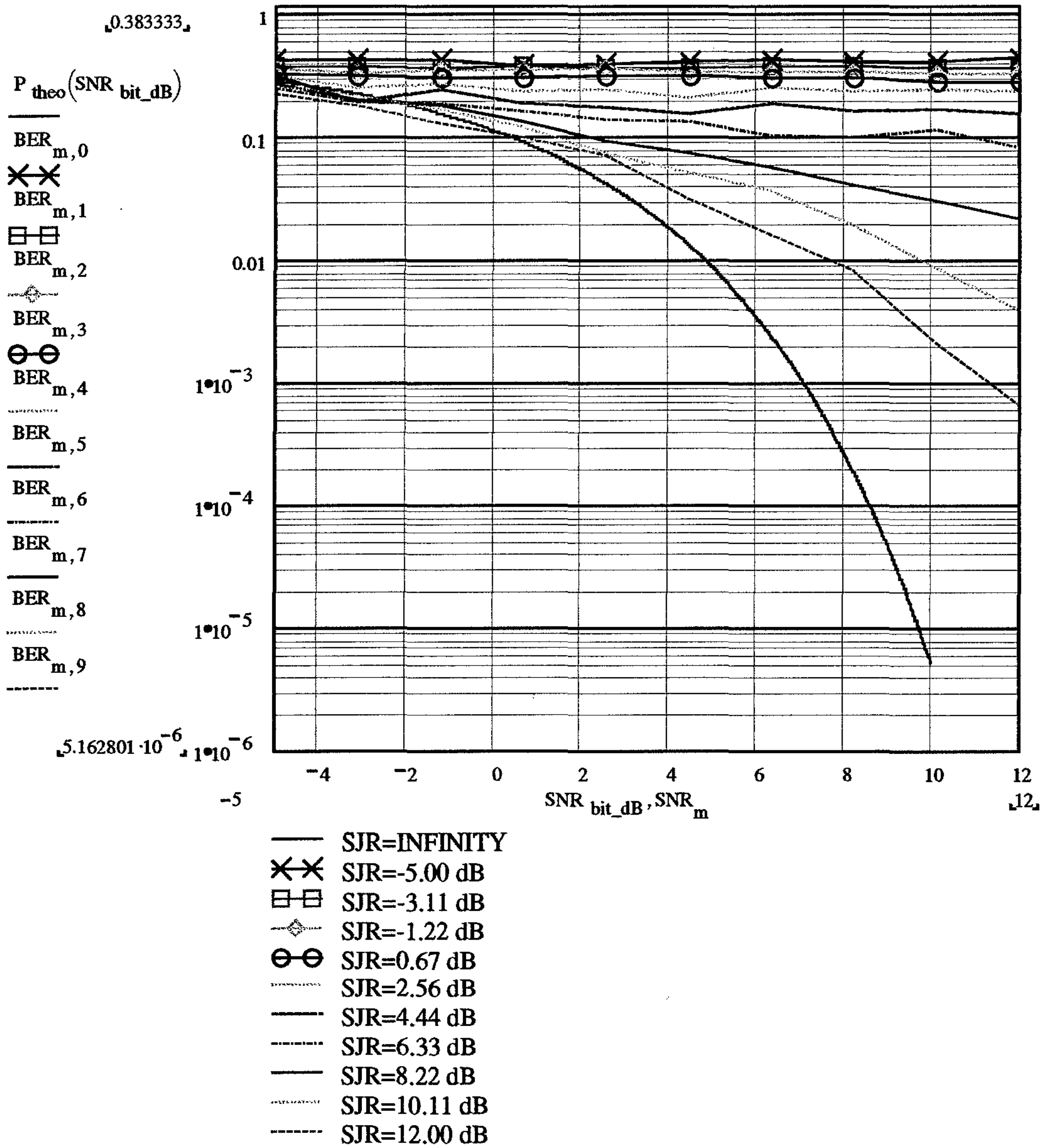


Figure 26. QPSK BER versus SNR with SJR as Parameter (Matlab Model Estimates)

The QPSK probability of bit error as a function of SJR, with SNR as a parameter starting at –5 dB and with 1.889-dB increments, is shown in Figure 27. The "step function" again represents the noise-free case, and the curves are seen to tend to the step function as the SNR increases.

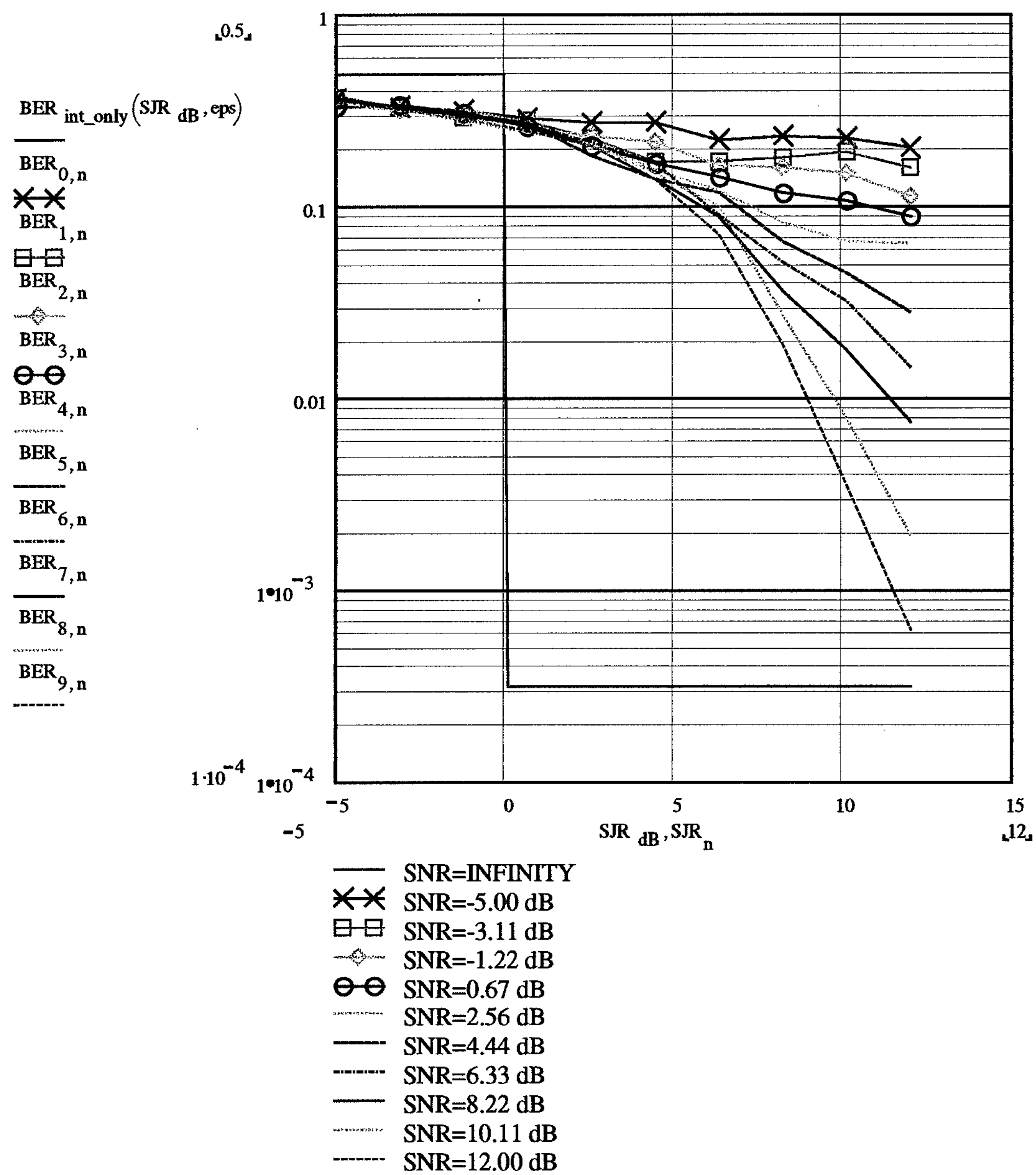


Figure 27. QPSK BER versus SJR with SNR as Parameter (Matlab Model Estimates)

From these figures, we see that co-channel interference acts like noise. When co-channel interference is high (SJR is low), the bit error probability is high and when it is low (SJR is high) the bit error probability is low provided SNR is also high.

From an examination of Figures 26 and 27, we see that we can maintain $P_b = 10^{-2}$ for SJR$\leq$ 12 dB when SNR$\geq$8.22 dB. Up to a point, further increasing SNR allows SJR to further decrease. A similar result is observed for BPSK, as can be seen from an examination of Figures 20 through 23. Hence, the importance of AWGN on the effect of co-channel interference cannot be overlooked.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The objective of this thesis was to model and simulate MPSK communication systems in the presence of additive white Gaussian noise and co-channel interference. The models were implemented as Simulink block-diagram models, using Simulink and Communications Toolbox blocks, as well as Matlab programs, using Matlab and specialized Communications Toolbox functions. Baseband equivalents of passband MPSK systems were modeled using the complex envelope concept. The results for the bit error probability were obtained for BPSK (2PSK) and QPSK (4PSK) modulation types with additive white Gaussian noise only and with the sum of additive white Gaussian noise and co-channel interference. Furthermore, the convergence (with increasing numbers of "transmitted" symbols) of bit error probability estimates obtained by simulation to the theoretical bit error probabilities has been verified for several cases of additive white Gaussian noise with different variances.

The simulation results For AWGN only are in general very close to the theoretical values and the estimates of the bit error rates converge to the theoretical values as the numbers of transmitted symbols increase. However, it has been observed that the Matlab model requires a 3 dB SNR correction for BPSK in noise case only. With this SNR correction, the simulation results for the bit error probability match the theoretical results and the results for the two implementations (SIMULINK and MATLAB) match each other. The BPSK and QPSK are both sensitive to co-channel interference almost to the same extent.

As a general rule, the sensitivity of both BPSK and QPSK to co-channel interference can be reduced by increasing the SNR. Hence, the importance of AWGN on the effect of co-channel interference cannot be overlooked. Of course, if the SJR becomes too small, then performance is poor regardless of the SNR.

B. RECOMMENDATIONS

This research can be continued to include forward error correction and adjacent channel interference in the developed models. Since only linear MPSK systems were considered, a future study may address transmitter non-linearity. Theoretical derivation of bit error probability can be obtained for MPSK with AWGN and interference which can be either phase locked to the signal or with some given distribution (such as Uniform or Gaussian) of the phase difference between the interference and the signal.

APPENDIX A. BATCH_RUN.M MATLAB PROGRAM FOR CONVERGENCE TEST OF SIMULATION USING BPSK

```
% This program runs multiple simulations for convergence of simulink%

clear

num_symbols = input('Enter the number of symbols [1000]: ');
if isempty(num_symbols), num_symbols = 1000; end



tic
sim('mpsk7_z',num_symbols);
disp('VAR 0.5 Done!')
toc

tic
sim('mpsk7_1',num_symbols);
disp('VAR 1 Done!')
toc

tic
sim('mpsk7_2',num_symbols);
disp('VAR 2 Done!')
toc

tic
sim('mpsk7_4',num_symbols);
disp('VAR 4 Done!')
toc

tic
sim('mpsk7_9',num_symbols);
disp('VAR 9 Done!')
toc

tic
sim('mpsk7_16',num_symbols);
disp('VAR 16 Done!')
toc
```

APPENDIX B. BER_PLOT.M MATLAB PROGRAM FOR PLOTTING THE FIGURES OF THE SIMULATION FOR EACH VARIANCES

```
% This loads two files and plots the result

clear

num_bits = input('Enter the number of bits [1000]: ');

if isempty(num_bits), num_bits = 1000; end

load d:\errnum1.dat

load d:\bitnum1.dat

plot(bitnum1, errnum1 ./ bitnum1, 'r-')

grid
xlabel('Bit Number')
ylabel('Bit Error Probability')
title('Bit Error Probability for variance 1')

[num_errors dummy] = size(errnum1);
disp('noise variance=1')
BER = num_errors / num_bits
```

APPENDIX C. PLOT_SIM_NOISE2 MATHCAD PROGRAM FOR RESULTS OF NOISE ONLY CASE BPSK USING SIMULINK MODEL

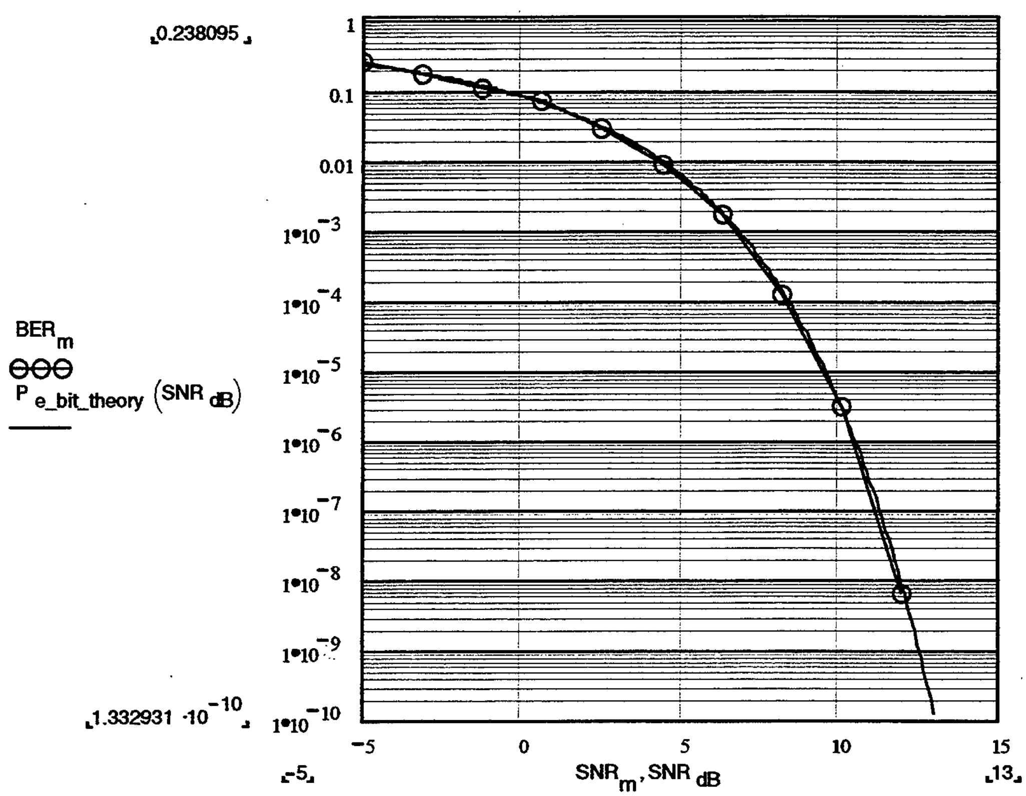
This program gives the probability of bit error vs. signal to noise ratio using outputs obtained from SIMULATION test for noise only BPSK.

BER := READPRN ("h:\veys\simnoise2.ber")

SNR := READPRN ("h:\veys\simnoise2.snr")

$SNR := SNR^{T}$ $\quad m := 0 .. \mathrm{rows}(SNR) - 1$

$$P_{e\_bit\_theory}(SNR_{dB}) := \frac{1}{2}\cdot\left(1 - \mathrm{erf}\left(\frac{1}{\sqrt{2}}\cdot\sqrt{2\cdot 10^{\frac{SNR_{dB}}{10}}}\right)\right) \qquad SNR_{dB} := -5, -4.9 .. 13$$



APPENDIX D. PLOT_MAT_NOISE2 MATHCAD PROGRAM FOR RESULTS OF NOISE ONLY CASE BPSK USING MATLAB PROGRAM

This program gives the probability of bit error vs. signal to noise ratio using outputs obtained from MATLAB test for noise only BPSK.

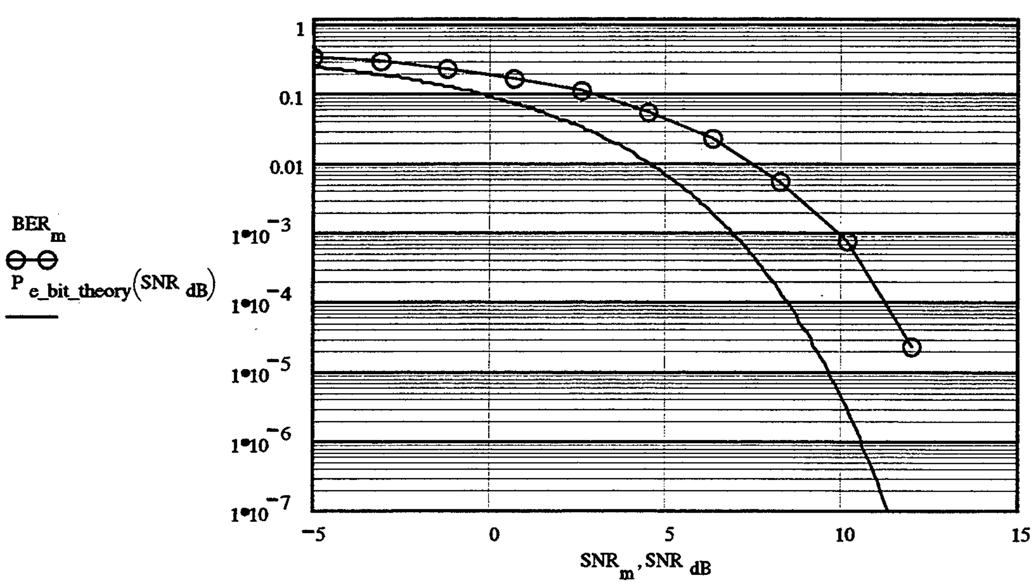
BER := READPRN("h:\veys\matnoise2.ber")

SNR := READPRN("h:\veys\matnoise2.snr")

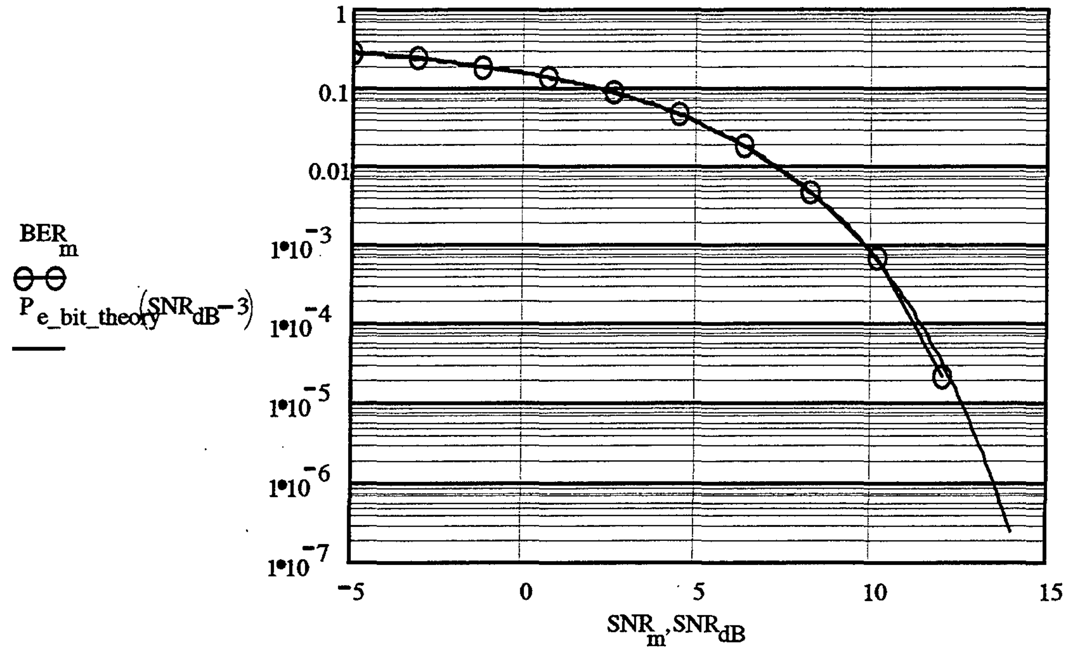
$SNR := SNR^T$

$m := 0 .. \mathrm{rows}(SNR) - 1$

$$P_{e\_bit\_theory}\left(SNR_{dB}\right) := \frac{1}{2} \cdot \left[1 - \mathrm{erf}\left[\frac{1}{\sqrt{2}} \cdot \sqrt{2 \cdot 10^{\frac{SNR_{dB}}{10}}}\right]\right]$$



plot with 3 db error



plot 3 dB corrected

APPENDIX E. PLOT_SIM_NOISE4 MATHCAD PROGRAM FOR RESULTS OF NOISE ONLY CASE QPSK USING SIMULINK MODEL

This program gives the probability of bit error vs. signal to noise ratio using outputs obtained from SIMULATION test for noise only QPSK.

$$M := 4$$

BER := READPRN("h:\veys\simnoise4.ber")

SNR := READPRN("h:\veys\simnoise4.snr")

$$SNR := SNR^T \qquad m := 0 .. \mathrm{rows}(SNR) - 1$$

$$BER := \frac{1}{2} \cdot \frac{M}{M-1} \cdot BER \qquad SNR_{bit\_dB\_sim_m} := SNR_m - 3$$

$$Q(x) := \frac{1}{2} \cdot \left(1 - \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

$$Q_f(M, SNR_{bit\_dB}) := Q\left(\sqrt{\frac{6}{M-1} \cdot \frac{\log(\sqrt{M})}{\log(2)} \cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)$$

$$P_{e\_bit\_QAM}(M, SNR_{bit\_dB}) := \frac{2}{1 + \frac{1}{\sqrt{M}}} \cdot Q_f(M, SNR_{bit\_dB}) \cdot \left[1 - \left(1 - \frac{1}{\sqrt{M}}\right) \cdot Q_f(M, SNR_{bit\_dB})\right]$$

$$P_{e\_bit\_ub}\left(M,SNR_{bit\_dB}\right) := \frac{M}{M-1}\cdot Q\left(\sqrt{\frac{\log(M)}{\log(2)}\cdot 2\cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\cdot \sin\left(\frac{\pi}{M}\right)\right)$$

$$\rho\left(\theta,M,SNR_{bit}\right) := \sqrt{10^{\frac{SNR_{bit}}{10}}\cdot\frac{\log(M)}{\log(2)}}\cdot\cos(\theta)$$

$$\Sigma_{erf}\left(\theta,M,SNR_{bit}\right) := 1+\mathrm{erf}\left(\rho\left(\theta,M,SNR_{bit}\right)\right)$$

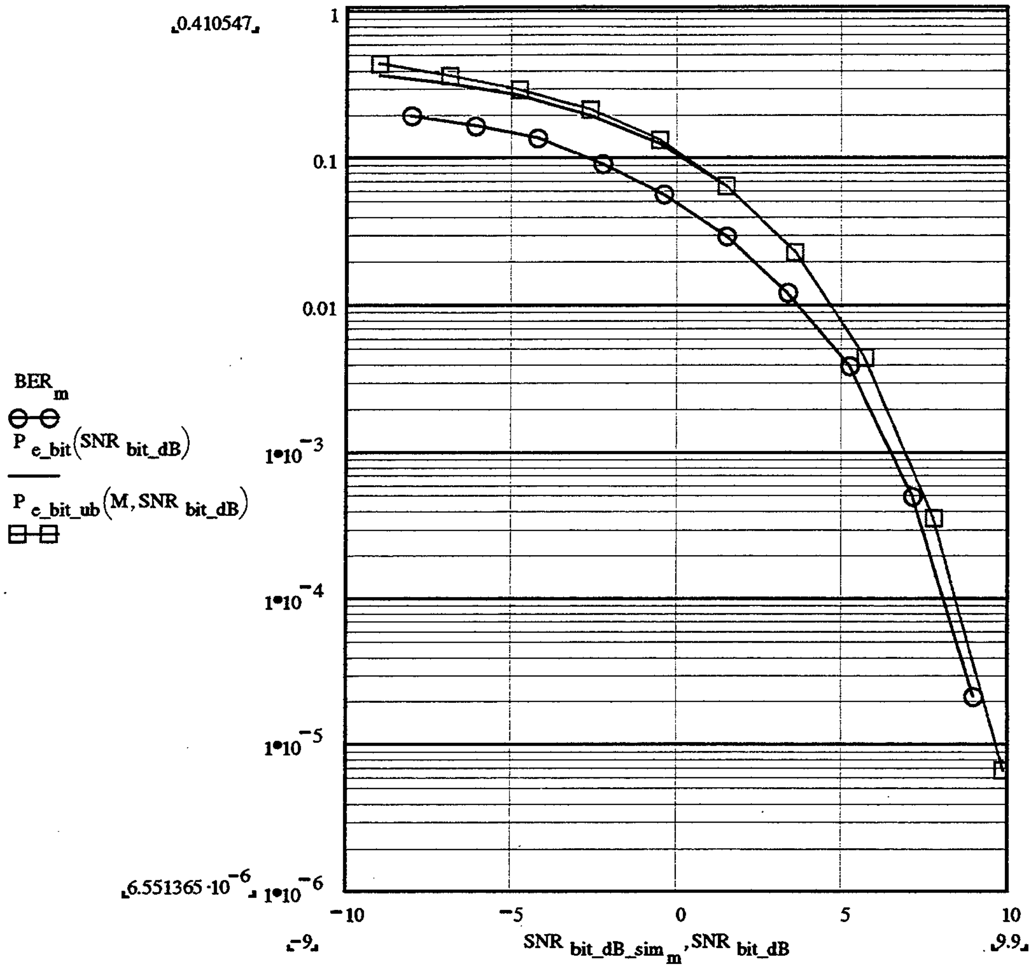
$$\Theta\left(\theta,M,SNR_{bit}\right) := \rho\left(\theta,M,SNR_{bit}\right)\cdot e^{\rho\left(\theta,M,SNR_{bit}\right)^2}\cdot\Sigma_{erf}\left(\theta,M,SNR_{bit}\right)$$

$$p_{\theta}\left(\theta,M,SNR_{bit}\right) := \frac{e^{-10^{\frac{SNR_{bit}}{10}}\cdot\frac{\log(M)}{\log(2)}}}{2\sqrt{\pi}}\cdot\left(\frac{1}{\sqrt{\pi}}+\Theta\left(\theta,M,SNR_{bit}\right)\right)$$

$$P_{e\_bit}\left(M,SNR_{bit}\right) := \frac{1}{2}\cdot\left[1-\left[\int_{-\frac{\pi}{M}}^{\frac{\pi}{M}} p_{\theta}\left(\theta,M,SNR_{bit}\right)d\theta\right]\right]$$

$$P_{e\_bit}\left(SNR_{bit\_dB}\right) := \frac{1}{2}\cdot\frac{M}{M-1}\cdot 2\cdot\left(Q\left(\sqrt{2\cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)\right)\cdot\left(1-\frac{1}{2}\cdot Q\left(\sqrt{2\cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)\right)$$

$$SNR_{bit\_dB} := -9,-6.9..10$$



0.410547
1
0.1
0.01
1•10^-3
1•10^-4
1•10^-5
6.551365·10^-6
1•10^-6
−10
−5
0
5
10
−9
9.9
SNR bit_dB_sim m, SNR bit_dB
BER m
P e_bit(SNR bit_dB)
P e_bit_ub(M, SNR bit_dB)

APPENDIX F. PLOT_MAT_NOISE4 MATHCAD PROGRAM FOR RESULTS OF NOISE ONLY CASE QPSK USING MATLAB PROGRAM

This program gives the probability of bit error vs. signal to noise ratio using outputs obtained from MATLAB test for noise only QPSK.

$M := 4$

BER := READPRN("h:\veys\matnoise4.ber")

SNR := READPRN("h:\veys\matnoise4.snr")

$SNR := SNR^T$

$m := 0..\,\mathrm{rows}(SNR) - 4$

$$Q(x) := \frac{1}{2}\cdot\left(1 - \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

$$Q_f\left(M, SNR_{bit\_dB}\right) := Q\left(\sqrt{\frac{6}{M-1}\cdot\frac{\log\left(\sqrt{M}\right)}{\log(2)}\cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)$$

$$P_{e\_bit\_QAM}\left(M, SNR_{bit\_dB}\right) := \frac{2}{1+\frac{1}{\sqrt{M}}}\cdot Q_f\left(M, SNR_{bit\_dB}\right)\cdot\left[1-\left(1-\frac{1}{\sqrt{M}}\right)\cdot Q_f\left(M, SNR_{bit\_dB}\right)\right]$$

$$P_{e\_bit\_ub}\left(M, SNR_{bit\_dB}\right) := \frac{M}{M-1} \cdot Q\left(\sqrt{\frac{\log(M)}{\log(2)} \cdot 2 \cdot 10^{\frac{SNR_{bit\_dB}}{10}}} \cdot \sin\left(\frac{\pi}{M}\right)\right)$$

$SNR_{bit\_dB\_sim_m} := SNR_m$ $\qquad$ $BER := BER$

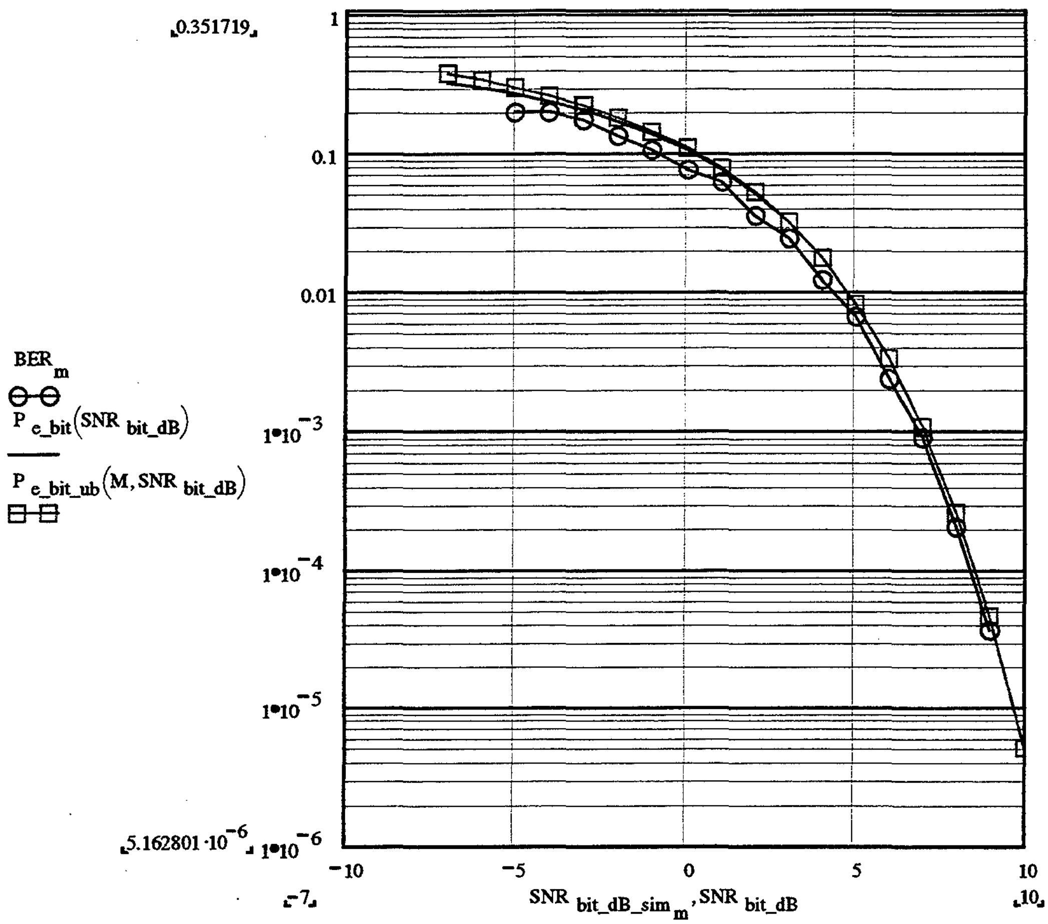
$$\rho\left(\theta, M, SNR_{bit}\right) := \sqrt{10^{\frac{SNR_{bit}}{10}} \cdot \frac{\log(M)}{\log(2)}} \cdot \cos(\theta)$$

$$\Sigma_{erf}\left(\theta, M, SNR_{bit}\right) := 1 + \operatorname{erf}\left(\rho\left(\theta, M, SNR_{bit}\right)\right)$$

$$\Theta\left(\theta, M, SNR_{bit}\right) := \rho\left(\theta, M, SNR_{bit}\right) \cdot e^{\rho\left(\theta, M, SNR_{bit}\right)^2} \cdot \Sigma_{erf}\left(\theta, M, SNR_{bit}\right)$$

$$p_\theta\left(\theta, M, SNR_{bit}\right) := \frac{e^{-10^{\frac{SNR_{bit}}{10}} \cdot \frac{\log(M)}{\log(2)}}}{2\sqrt{\pi}} \cdot \left(\frac{1}{\sqrt{\pi}} + \Theta\left(\theta, M, SNR_{bit}\right)\right)$$

$$P_{e\_bit}\left(M, SNR_{bit}\right) := \frac{1}{2} \cdot \left[1 - \left[\int_{-\frac{\pi}{M}}^{\frac{\pi}{M}} p_\theta\left(\theta, M, SNR_{bit}\right) d\theta\right]\right]$$


0.351719
1
0.1
0.01
1•10^-3
1•10^-4
1•10^-5
5.162801·10^-6
1•10^-6
BER_m
P e_bit(SNR bit_dB)
P e_bit_ub(M, SNR bit_dB)
−10
−5
0
5
10
-7
SNR bit_dB_sim_m, SNR bit_dB
10

APPENDIX G. PLOT_SIM_NOISE&INT2 MATHCAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE BPSK USING SIMULINK MODEL

This program gives the probability of bit error vs. signal to noise ratio and probability of bit error vs.signal to jamming ratio using outputs obtained from SIMULATION test for BPSK.

BER := READPRN ("h:\veys\simnoint2.ber")

SNR := READPRN ("h:\veys\simnoint2.snr")

SJR := READPRN ("h:\veys\simnoint2.sjr")

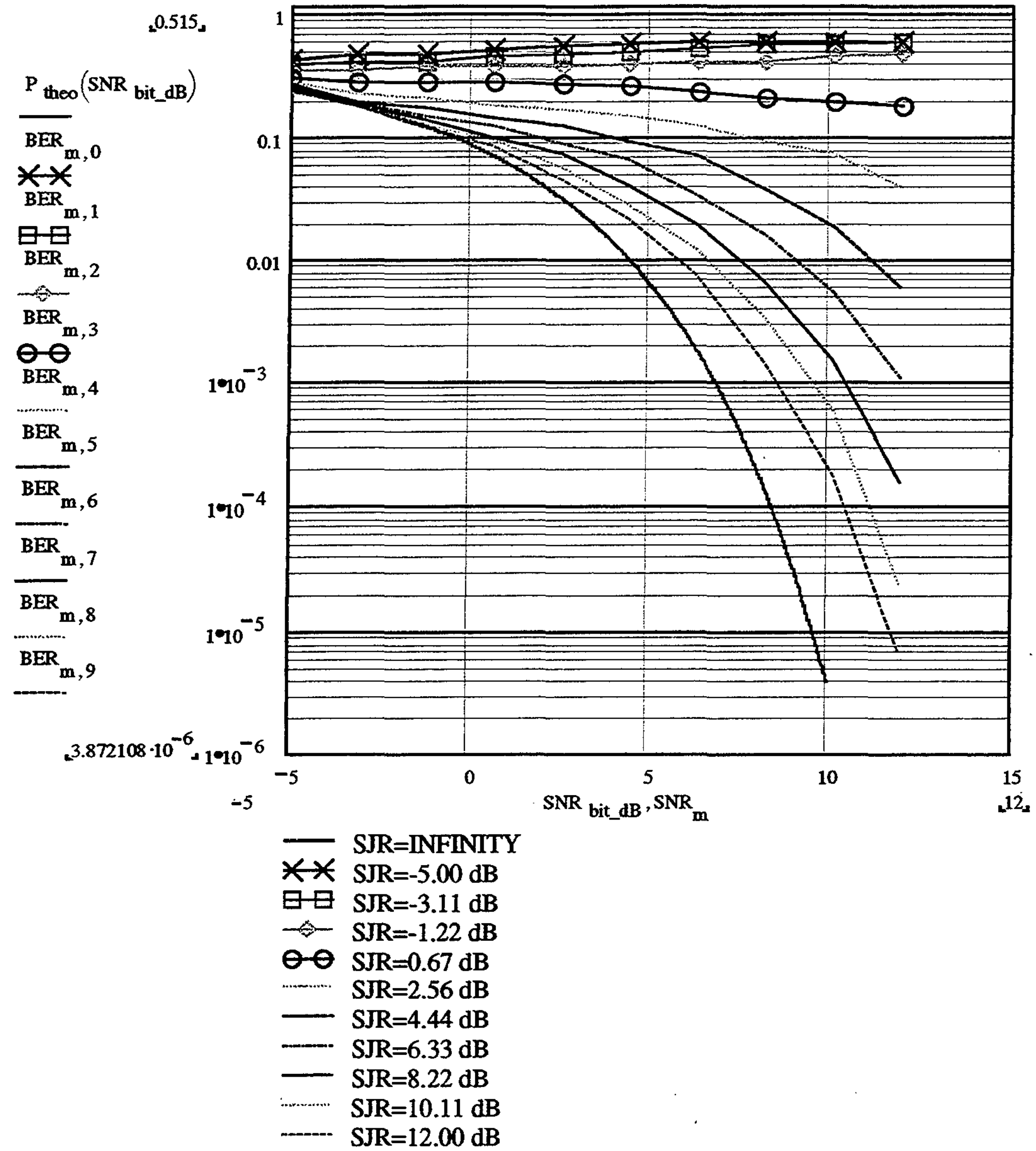
$SNR := SNR^T$ $\qquad$ $SJR := SJR^T$

$m := 0 .. \mathrm{rows}(SNR) - 1$

$n := 0 .. \mathrm{rows}(SJR) - 1$

$$P_{theo}(SNR_{dB}) := \frac{1}{2} \cdot \left[1 - \mathrm{erf}\left[\frac{1}{\sqrt{2}} \cdot \sqrt{2 \cdot 10^{\frac{SNR_{dB}}{10}}} \right] \right]$$

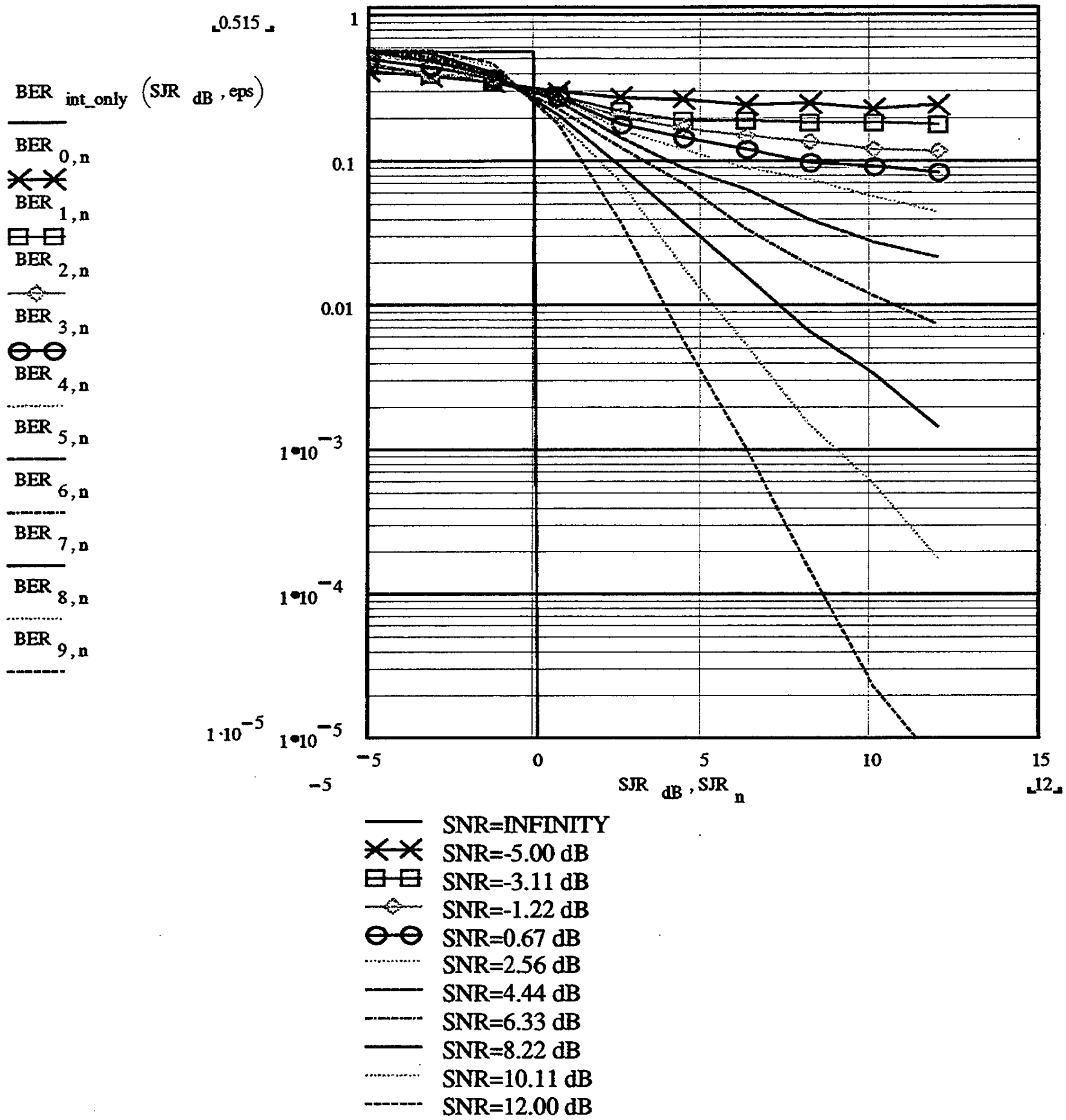
$SNR_{dB} := -5 .. 12$



$eps := 10^{-5}$

$BER_{int\_only}(SJR_{dB}, eps) := if(SJR_{dB} < 0, 0.5, eps)$

$SJR_{dB} := -5, -4.9 .. 12$



0.515
BER int_only (SJR dB, eps)
BER 0,n
BER 1,n
BER 2,n
BER 3,n
BER 4,n
BER 5,n
BER 6,n
BER 7,n
BER 8,n
BER 9,n
1
0.1
0.01
1•10−3
1•10−4
1·10−5
1•10−5
−5
0
5
10
15
−5
SJR dB, SJR n
12
SNR=INFINITY
SNR=-5.00 dB
SNR=-3.11 dB
SNR=-1.22 dB
SNR=0.67 dB
SNR=2.56 dB
SNR=4.44 dB
SNR=6.33 dB
SNR=8.22 dB
SNR=10.11 dB
SNR=12.00 dB

APPENDIX H. PLOT_MAT_NOISE&INT2 MATHCAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE BPSK USING MATLAB PROGRAM

This program gives the probability of bit error vs. signal to noise ratio and probability of bit error vs.signal to jamming ratio using outputs obtained from MATLAB test for BPSK.

BER := READPRN ("h:\veys\matnoint2.ber")

SNR := READPRN ("h:\veys\matnoint2.snr")

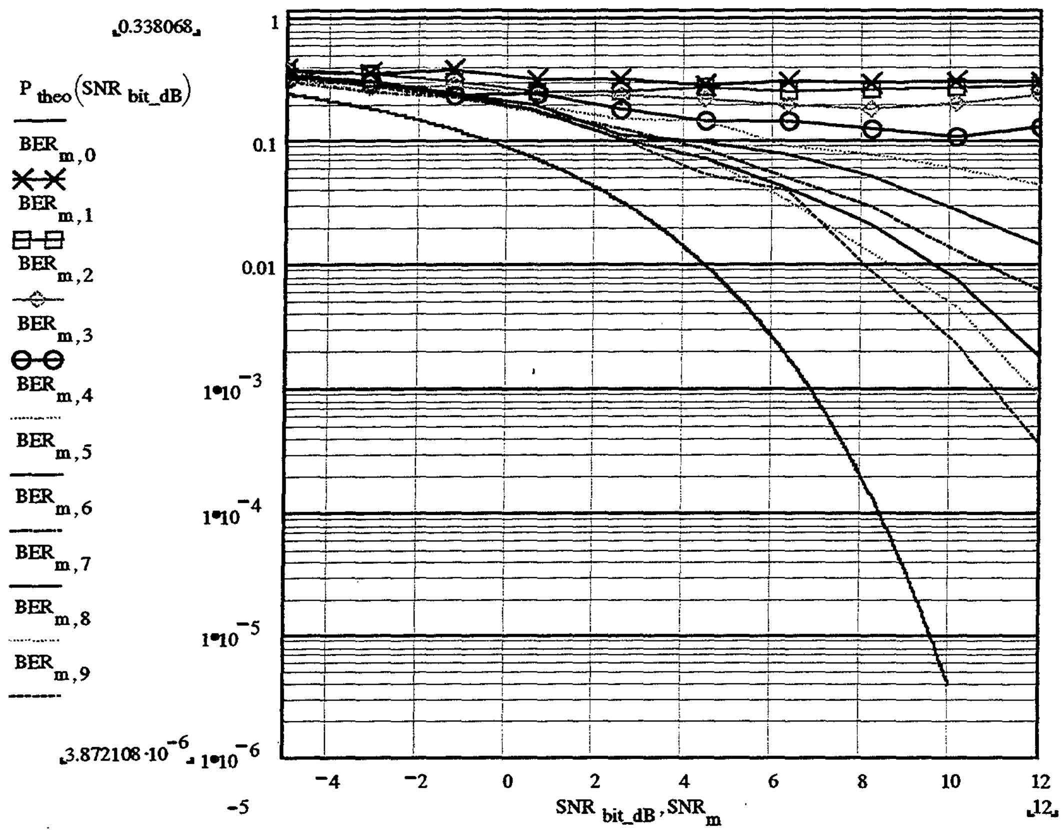
SJR := READPRN ("h:\veys\matnoint2.sjr")

$SNR := SNR^{T}$ $\qquad$ $SJR := SJR^{T}$

$m := 0 .. \mathrm{rows}(SNR) - 1$

$n := 0 .. \mathrm{rows}(SJR) - 1$

$$P_{theo}\left(SNR_{dB}\right) := \frac{1}{2}\cdot\left[1 - \mathrm{erf}\left[\frac{1}{\sqrt{2}}\cdot\sqrt{2\cdot 10^{\frac{SNR_{dB}}{10}}}\right]\right] \qquad SNR_{dB} := -5 .. 12$$

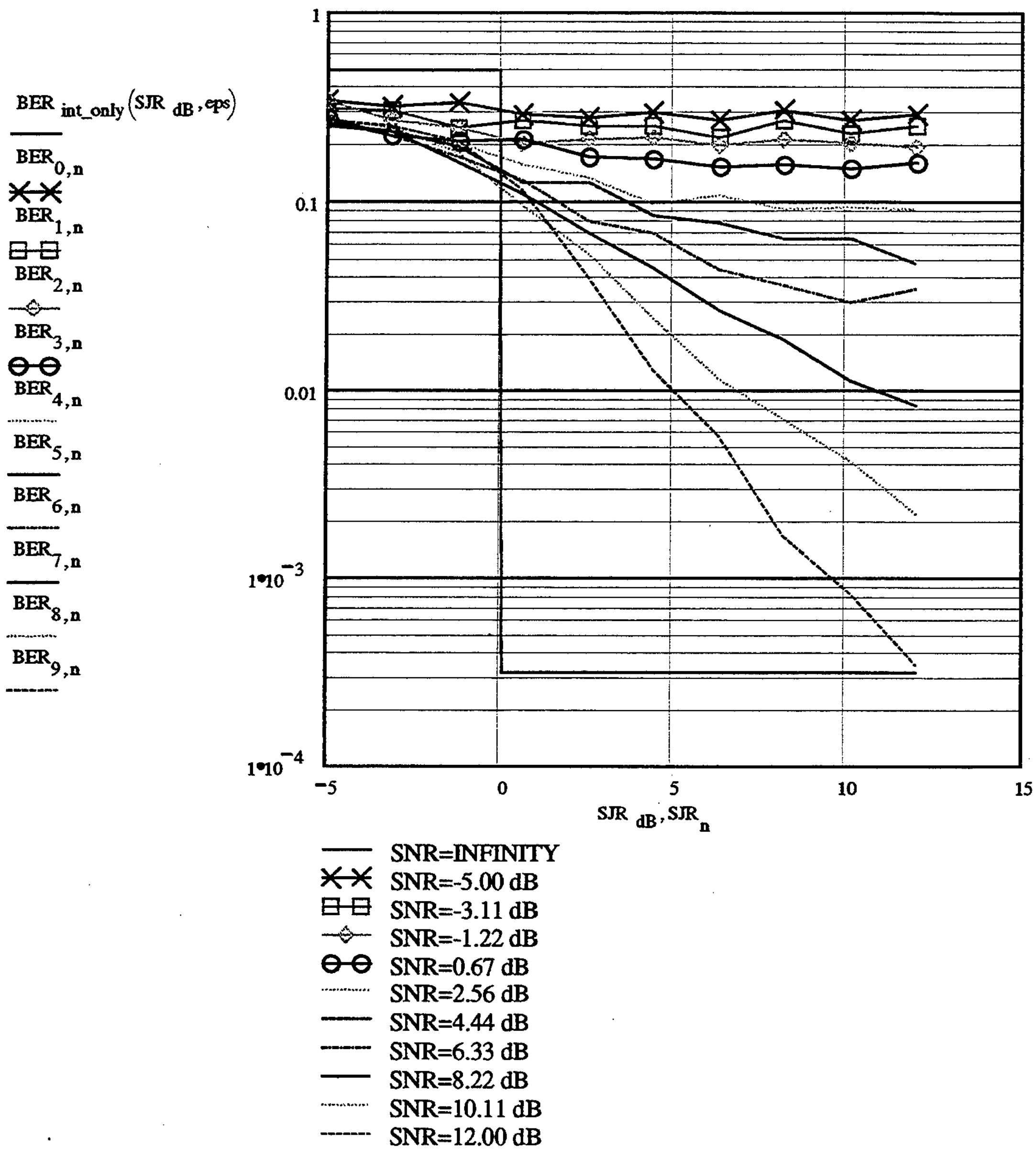


SJR=INFINITY
SJR=-5.00 dB
SJR=-3.11 dB
SJR=-1.22 dB
SJR=0.67 dB
SJR=2.56 dB
SJR=4.44 dB
SJR=6.33 dB
SJR=8.22 dB
SJR=10.11 dB
SJR=12.00 dB

$eps := 10^{-3.5}$

$BER_{int\_only}(SJR_{dB}, eps) := if(SJR_{dB} < 0, 0.5, eps)$

$SJR_{dB} := -5, -4.9 .. 12$



APPENDIX I. PLOT_SIM_NOISE&INT4 MATHCAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE QPSK USING SIMULINK MODEL

This program gives the probability of bit error vs. signal to noise ratio and probability of bit error vs.signal to jamming ratio using outputs obtained from SIMULATION test for QPSK.

$M := 4$

BER := READPRN ("h:\veys\simnoint4.ber")

SNR := READPRN ("h:\veys\simnoint4.snr")

SJR := READPRN ("h:\veys\simnoint4.sjr")

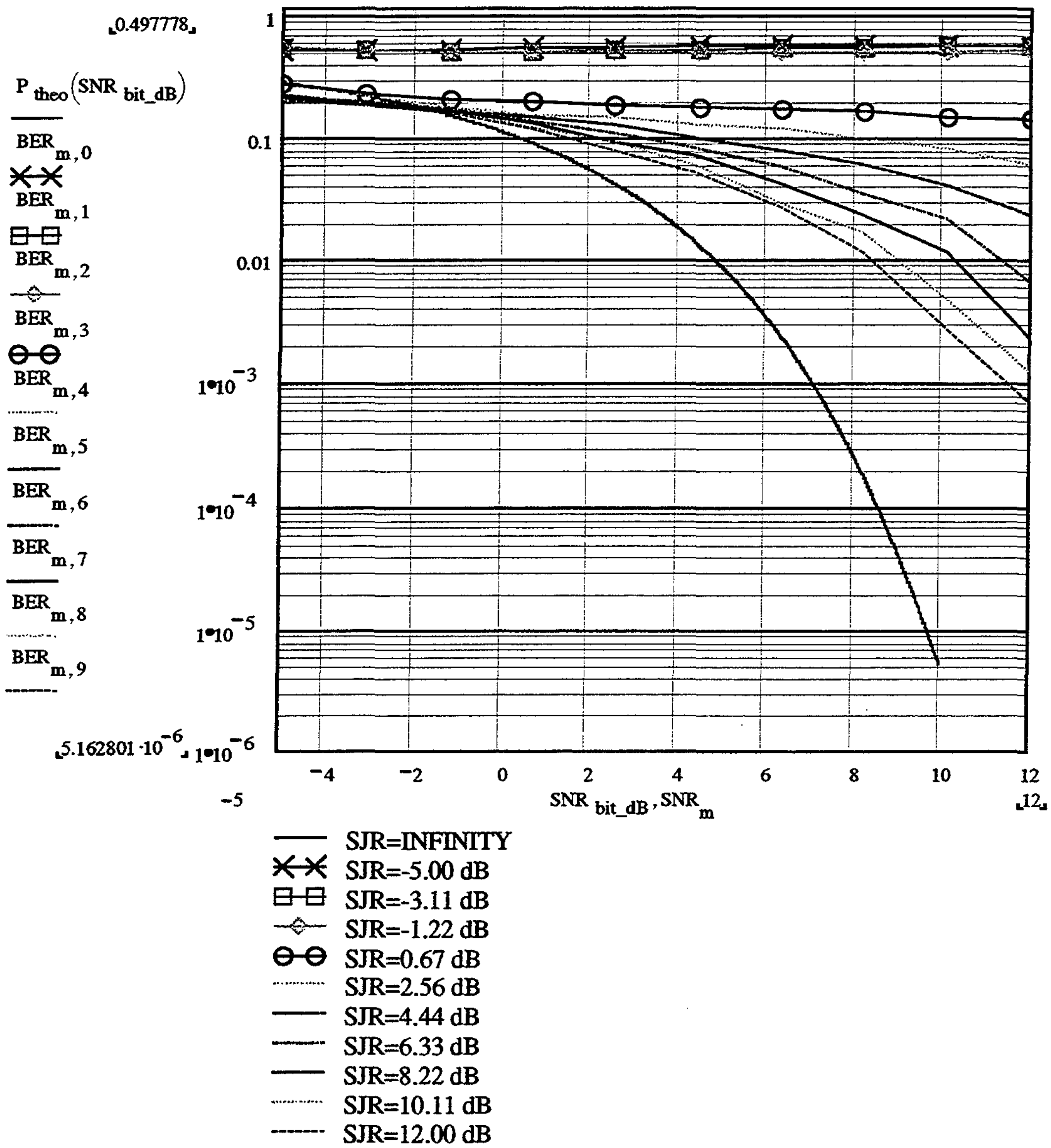
$SNR := SNR^T - 3 \qquad SJR := SJR^T \qquad BER := \frac{1}{2} \cdot \frac{M}{M-1} \cdot BER$

$m := 0 .. \text{rows}(SNR) - 1$

$n := 0 .. \text{rows}(SJR) - 1$

$$Q(x) := \frac{1}{2} \cdot \left(1 - \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

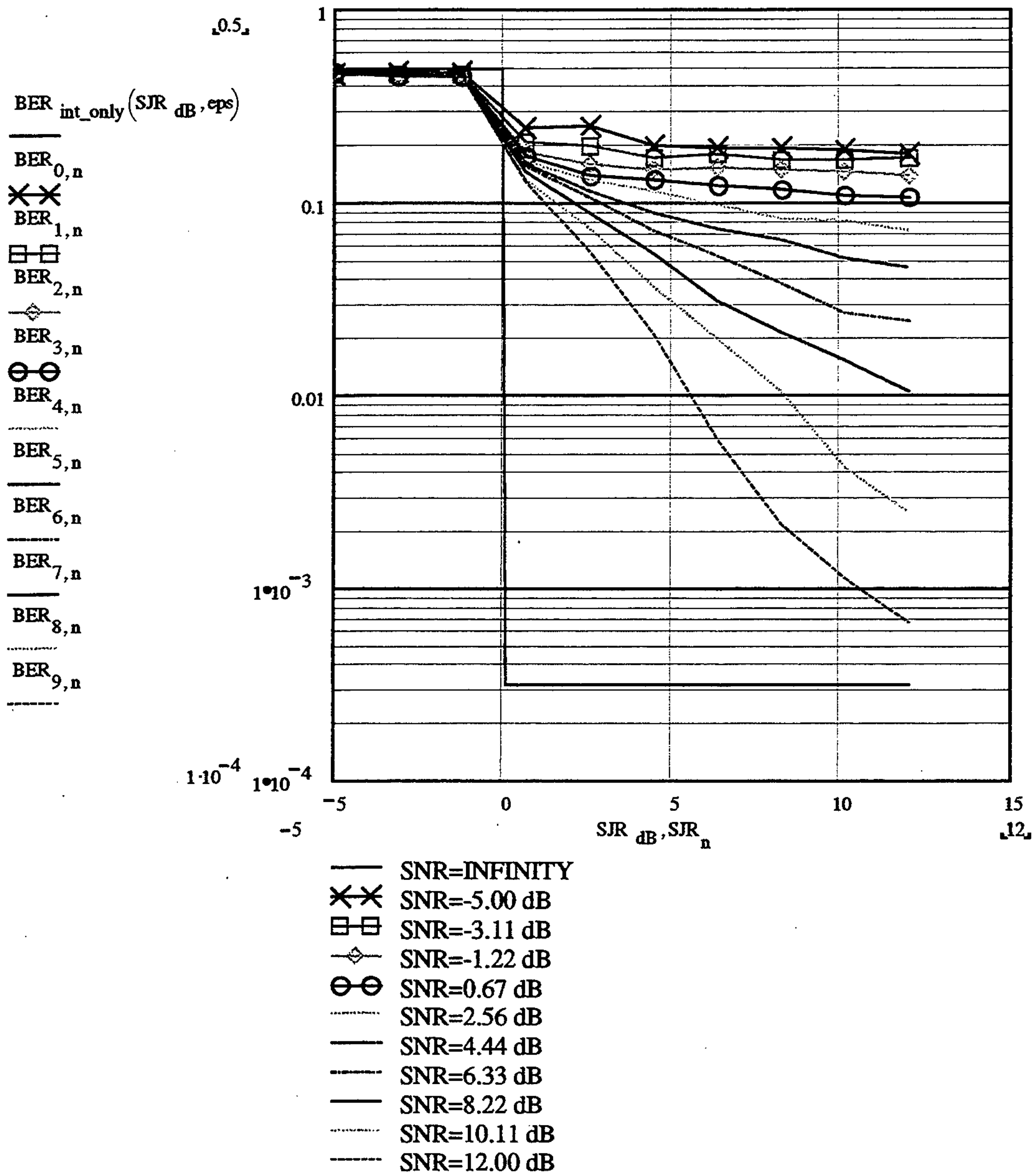
$$P_{theo}\left(SNR_{bit\_dB}\right) := \frac{1}{2} \cdot \frac{M}{M-1} \cdot 2 \cdot \left(Q\left(\sqrt{2 \cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)\right) \cdot \left(1 - \frac{1}{2} \cdot Q\left(\sqrt{2 \cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)\right)$$



$$eps := 10^{-3.5}$$

$$BER_{int\_only}\left(SJR_{dB}, eps\right) := if\left(SJR_{dB} < 0, 0.5, eps\right)$$

$$SJR_{dB} := -5, -4.9 .. 12$$


0.5
BER int_only(SJR dB, eps)
BER0,n
BER1,n
BER2,n
BER3,n
BER4,n
BER5,n
BER6,n
BER7,n
BER8,n
BER9,n
1
0.1
0.01
1•10−3
1·10−4
1•10−4
−5
0
5
10
15
-5
SJR dB, SJRn
12
SNR=INFINITY
SNR=-5.00 dB
SNR=-3.11 dB
SNR=-1.22 dB
SNR=0.67 dB
SNR=2.56 dB
SNR=4.44 dB
SNR=6.33 dB
SNR=8.22 dB
SNR=10.11 dB
SNR=12.00 dB

APPENDIX J. PLOT_MAT_NOISE&INT4 MATHCAD PROGRAM FOR RESULTS OF NOISE AND INTERFERENCE CASE QPSK USING MATLAB PROGRAM

This program gives the probability of bit error vs. signal to noise ratio and probability of bit error vs.signal to jamming ratio using outputs obtained from MATLAB test for QPSK.

$M := 4$

BER := READPRN ("h:\veys\matnoint4.ber")

SNR := READPRN ("h:\veys\matnoint4.snr")

SJR := READPRN ("h:\veys\matnoint4.sjr")

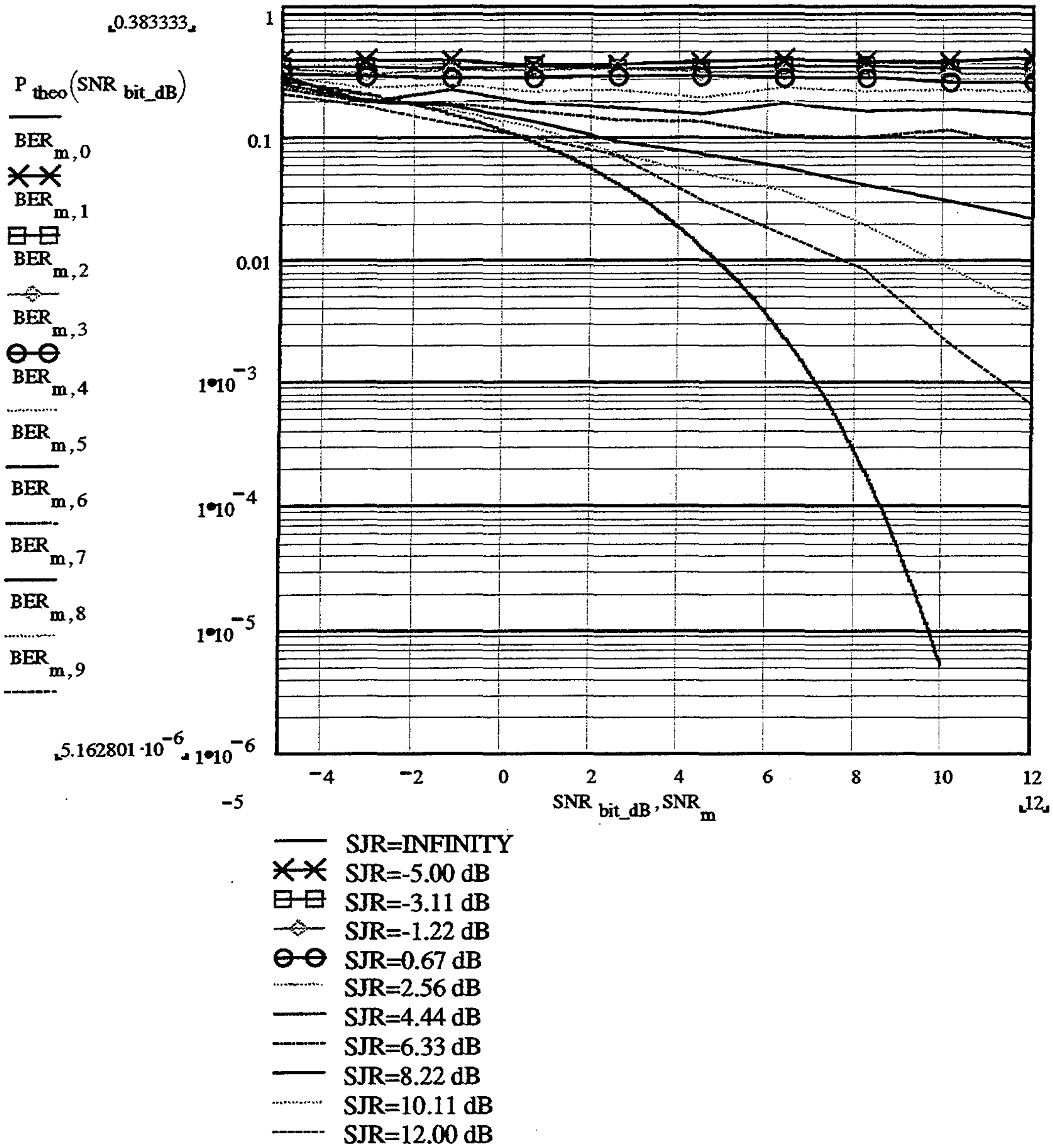
$SNR := SNR^T \quad SJR := SJR^T \qquad BER := BER$

$m := 0 .. \text{rows}(SNR) - 1$

$n := 0 .. \text{rows}(SJR) - 1$

$$Q(x) := \frac{1}{2}\cdot\left(1 - \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

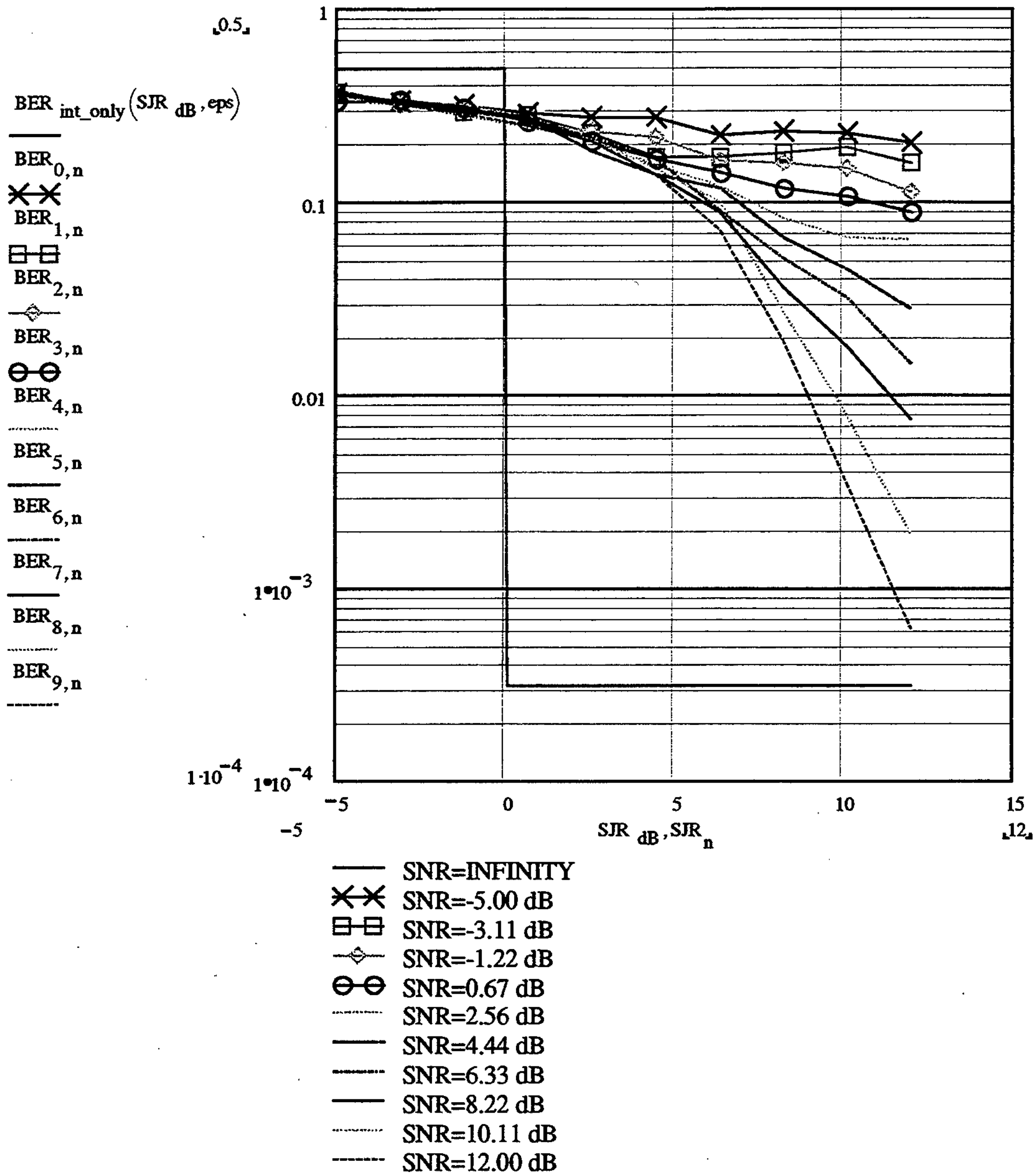
$$P_{theo}\left(SNR_{bit\_dB}\right) := \frac{1}{2}\cdot\frac{M}{M-1}\cdot 2\cdot\left(Q\left(\sqrt{2\cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)\right)\cdot\left(1-\frac{1}{2}\cdot Q\left(\sqrt{2\cdot 10^{\frac{SNR_{bit\_dB}}{10}}}\right)\right)$$



$eps := 10^{-3.5}$

$BER_{int\_only}\left(SJR_{dB}, eps\right) := if\left(SJR_{dB} < 0, 0.5, eps\right)$

$SJR_{dB} := -5, -4.9 .. 12$



0.5
BER int_only(SJR dB, eps)
BER0,n
BER1,n
BER2,n
BER3,n
BER4,n
BER5,n
BER6,n
BER7,n
BER8,n
BER9,n
1
0.1
0.01
1·10^-3
1·10^-4
1·10^-4
-5
0
5
10
15
-5
SJR dB, SJRn
12
SNR=INFINITY
SNR=-5.00 dB
SNR=-3.11 dB
SNR=-1.22 dB
SNR=0.67 dB
SNR=2.56 dB
SNR=4.44 dB
SNR=6.33 dB
SNR=8.22 dB
SNR=10.11 dB
SNR=12.00 dB

APPENDIX K. PREPAPE.M MATLAB PROGRAM TO ENTER THE SYSTEM PARAMETERS TO BE USED BY THE SIMUINK MODEL

```
%%% This prepares the data file for simulink runs
clear

noise_only = menu(' Select: ',...
            'NOISE',...
            'NOISE and INTERFERENCE ');

num_levels  = input('Enter the number of frequencies M [2]: ');
if isempty(num_levels), num_levels = 2; end

T_sym       = input('Enter the symbol duration T [1]: ');
if isempty(T_sym), T_sym = 1; end

oversampling = input('Enter the oversampling factor [2]: ');
if isempty(oversampling), oversampling = 2; end

min_SNR = input('Enter the MIN Signal to Noise ratio[-5 dB]: ');
if isempty(min_SNR), min_SNR = -5; end

max_SNR = input('Enter the MAX Signal to Noise ratio [12 dB]: ');
if isempty(max_SNR), max_SNR = 12; end
if noise_only ~= 1
    min_SJR = input('Enter the MIN Signal to Interference ratio [-5 dB]: ');
    if isempty(min_SJR), min_SJR = -5; end
    max_SJR = input('Enter the MAX Signal to Interference ratio[12 dB]: ');
  if isempty(max_SJR), max_SJR = 12; end
end
if min_SNR == max_SNR
  num_noise = 1;
else
    num_noise = input('Enter the number of values for SNR [10]: ');
  if isempty(num_noise), num_noise = 10; end
end

if noise_only ~= 1
  if min_SJR == max_SJR
    num_jam = 1;
  else
        num_jam = input('Enter the number of values for SJR [10]: ');
    if isempty(num_jam), num_jam = 10; end
  end
```

```
else
  num_jam = 1;
end

min_errors = input('Enter the min number of errors acceptable [100]: ');
if isempty(min_errors), min_errors = 100; end

error_factor = input('Enter the factor multiplying the number of errors [2]: ');
if isempty(error_factor), error_factor = 2; end

initial_num_symbols = error_factor*min_errors;

max_randint = input('Enter the maximum size of the random integer arrays [10^6]: ');
if isempty(max_randint), max_randint = 10^6; end

file_name = input('Enter the file name to save data [no ext]: ','s');

save sun_data
```

APPENDIX L. MPSK.M MATLAB PROGRAM TO START SIMULATION

```
%%% This runs MPSK co-channel interference with additive noise
clear

load sun_data  %This loads all the input data

delta_t     = T_sym/(2*oversampling)
seeds       = randint(3,1,1000);
signal_seed = seeds(1);
noise_seed  = seeds(2);
interf_seed = seeds(3);
initial_num_symbols = error_factor*min_errors;

tic

if num_noise > 1
  delta_SNR = (max_SNR - min_SNR) / (num_noise -1);
else
  delta_SNR = 0;
end
SNR = min_SNR + [0:num_noise - 1]*delta_SNR;
%noise_var_vect  = T_sym/(2*delta_t) .* 10 .^ (-SNR/10);
noise_var_vect  = 10 .^ (-SNR/10);


if noise_only ~= 1
  if num_jam > 1
    delta_SJR = (max_SJR - min_SJR) / (num_jam -1);
  else
    delta_SJR = 0;
  end
  SJR = min_SJR + [0:num_jam  - 1]*delta_SJR;
  interf_gain_vect = 10 .^ (-SJR/20);
else
  SJR = - 100;          % There is no Jamming so the SJR in dB is -infinity
  interf_gain_vect = 0;
end

BER = zeros(num_noise,num_jam);
total_symbols = initial_num_symbols;
for noise_case = 1:num_noise
  noise_var = noise_var_vect(noise_case);
  for jam_case = 1:num_jam
     enough_errors = 0;
```

```
      num_err      = 0;
      num_symbols  = initial_num_symbols;
      rand_int     = min([num_symbols max_randint]);
      total_symbols = num_symbols

      while enough_errors ~= 1
        interf_gain = interf_gain_vect(jam_case);
        rand_int=min([10^6 num_symbols]);
        clear error_number
            sim('pskco_bm',num_symbols)
            [new_errors err_cols] = size(error_number);
            num_err = num_err + new_errors;
            if num_err == 0
           num_symbols  = num_symbols*min_errors;
           rand_int     = min([num_symbols max_randint]);
           total_symbols = total_symbols + num_symbols;
            elseif (num_err > 0 & num_err < min_errors)
           num_symbols  = (min_errors - num_err)*ceil(total_symbols/num_err);
           total_symbols = total_symbols + num_symbols;
        else
           enough_errors = 1;
            end
      end

      number_of_errors(noise_case,jam_case)  = num_err
      number_of_symbols(noise_case,jam_case) = total_symbols
      BER(noise_case,jam_case)               = num_err / total_symbols
    end
    eval([' save ' file_name '.snr SNR -ascii']);
    eval([' save ' file_name '.sjr SJR -ascii']);
    eval([' save ' file_name '.ber BER -ascii']);
    eval([' save ' file_name '.ner number_of_errors -ascii']);
    eval([' save ' file_name '.nsy number_of_symbols -ascii'])
end
toc
```

APPENDIX M. MPSK1.M MATLAB PROGRAM FOR USING MATLAB PROGRAM OPTION VICE SIMULINK MODEL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Matlab Version for MPSK                             %%%
%%% Dr. Jovan Lebaric                                   %%%
%%% June 14, 1998                                       %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% User Input %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

j = sqrt(-1);

noise_only = menu(' Select: ',...
           ' NOISE',...
           ' NOISE and INTERFERENCE ');M  = input('Enter the number of phases
(power of 2) M [2]: ');
if isempty(M), M = 2; end

T_sym = input('Enter the symbol duration T [1]: ');
if isempty(T_sym), T_sym = 1; end

oversampling = input('Enter the oversampling factor [2]: ');
if isempty(oversampling), oversampling = 2; end

min_SNR_bit = input('Enter the Minimum BIT Signal to Noise ratio in dB[-5 dB]: ');
if isempty(min_SNR_bit), min_SNR_bit = -5; end

max_SNR_bit = input('Enter the Maximum BIT Signal to Noise ratio in dB [+12 dB]: ');
if isempty(max_SNR_bit), max_SNR_bit = 12; end

if noise_only ~= 1
    min_SJR_bit = input('Enter the Minimum BIT Signal to Interference ratio in dB [-5
dB]: ');
  if isempty(min_SJR_bit), min_SJR_bit = -5; end
    max_SJR_bit = input('Enter the Maximum BIT Signal to Interference ratio in dB
[+12 dB]: ');
  if isempty(max_SJR_bit), max_SJR_bit = 12; end
```

```
end

if max_SNR_bit == min_SNR_bit
  num_noise = 1;
else
    num_noise = input('Enter the number of values for SNR [18]: ');
  if isempty(num_noise), num_noise = 10; end
end

if noise_only ~= 1
  if min_SJR_bit == max_SJR_bit
    num_jam = 1;
  else
       num_jam = input('Enter the number of values for SJR [10]: ');
    if isempty(num_jam), num_jam = 10; end
  end
else
  num_jam = 1;
end

min_errors = input('Enter the min number of errors acceptable [100]: ');
if isempty(min_errors), min_errors = 100; end

error_factor = input('Enter the factor multiplying the number of errors [2]: ');
if isempty(error_factor), error_factor = 2; end

initial_num_symbols = error_factor*min_errors;

max_randint = input('Enter the maximum size of the random integer arrays [10^6]: ');
if isempty(max_randint), max_randint = 10^6; end

file_name = input('Enter the file name to save data [no ext]: ','s');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Parameters for MPSK %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Symbol Frequency %%%
Fd = 1/T_sym;

%%% Sampling Frequency %%%
Fs = oversampling*Fd;

%%% Sampling Interval %%%
delta_t = 1/Fs;
```

```
%%% Seeds for Random Integers %%%
seeds      = randint(2,1,1000);
signal_seed = seeds(1);
interf_seed = seeds(2);

initial_num_symbols = error_factor*min_errors;

tic

%%% SNR Array %%%
if num_noise > 1
  delta_SNR_bit = (max_SNR_bit - min_SNR_bit) / (num_noise -1);
else
  delta_SNR_bit = 0;
end
SNR_bit = min_SNR_bit + [0:num_noise - 1]*delta_SNR_bit;
SNR_symbol = SNR_bit + 10*log10(log(M)/log(2));

%%% Noise Variance Array %%%%
noise_var_vect  = 10 .^ (-SNR_symbol/10);
%noise_var_vect  = T_sym/ (2*delta_t) * 10 .^ (-SNR_symbol/10);

%%% SJR Array %%%
if noise_only ~= 1

  if num_jam > 1
    delta_SJR_bit = (max_SJR_bit - min_SJR_bit) / (num_jam -1);
  else
    delta_SJR_bit = 0;
  end

  SJR_bit = min_SJR_bit + [0:num_jam  - 1]*delta_SJR_bit;
  SJR_symbol = SJR_bit + 10*log10(log(M)/log(2));
  interf_gain_vect = 10 .^ (-SJR_symbol/20);

else

  SJR_symbol = 100;
  SJR_bit   = 100;  % There is no Jamming so the SJR is large (infinity)

end

BER          = zeros(num_noise,num_jam);
total_symbols = initial_num_symbols;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Noise and Interference Loops %%%

for noise_case = 1:num_noise

  noise_var = noise_var_vect(noise_case);

  for jam_case = 1:num_jam

     enough_errors = 0;
     num_err       = 0;
     num_symbols   = initial_num_symbols;
     rand_int      = min([num_symbols max_randint]);
     total_symbols = num_symbols

     while enough_errors ~= 1


        rand_int       = min([10^6 num_symbols]);
        input_sequence = randint(rand_int, 1,[0 M-1], signal_seed);

        %%%%%%%%%%%%%%%%%%%
        %%% Modulation %%%%
        mod_MPSK =  dmodce(input_sequence, Fd, Fs, 'psk', M);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%% Additive Gaussian Noise %%%
        randn('state',sum(100*clock));
        complex_noise = randn(rand_int*oversampling,1)*sqrt(noise_var);
       %randn('state',sum(200*clock));
       %complex_noise = complex_noise +
j*randn(rand_int*oversampling,1)*sqrt(noise_var);

        mod_MPSK = mod_MPSK + complex_noise;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%% Additive Interference %%%
        if noise_only ~= 1
          interf_gain   = interf_gain_vect(jam_case);
          mod_MPSK = mod_MPSK + interf_gain*randint(rand_int*oversampling, 1,[0
M-1], interf_seed);
        end

        %%%%%%%%%%%%%%%%%%%%
```

```
    %%% Demodulation %%%
    demod_MPSK = ddemodce(mod_MPSK, Fd,Fs,'psk',M);

    %%%%%%%%%%%%%
    %%% Errors %%%

    [symbols_in_error dummy_col] = find(demod_MPSK - input_sequence);
    [new_errors dummy_col] = size(symbols_in_error);
    num_err = num_err + new_errors;

       if num_err == 0
      num_symbols   = num_symbols*min_errors;
      rand_int      = min([num_symbols max_randint]);
      total_symbols = total_symbols + num_symbols;
       elseif (num_err > 0 & num_err < min_errors)
      num_symbols   = (min_errors - num_err)*ceil(total_symbols/num_err);
      total_symbols = total_symbols + num_symbols;
    else
      enough_errors = 1;
    end

  end

  number_of_errors(noise_case,jam_case)  = num_err
  number_of_symbols(noise_case,jam_case) = total_symbols
  BER(noise_case,jam_case)            = 0.5 * M/(M-1) * num_err / total_symbols

 end

 eval([' save ' file_name '.snr SNR_bit -ascii']);
 eval([' save ' file_name '.sjr SJR_bit -ascii']);
 eval([' save ' file_name '.ber BER -ascii']);
 eval([' save ' file_name '.ner number_of_errors -ascii']);
 eval([' save ' file_name '.nsy number_of_symbols -ascii']);

end

toc
```

LIST OF REFERENCES

1. Communications Toolbox for Use with MATLAB and SIMULINK, The MATHWORKS Inc.

2. Hwei P.Hsu, Schaum's Outlines, Analog and Digital Communications, McGraw-Hill, 1993.

3. Bernard Sklar, Digital Communications Fundamental and Applications, Prentice Hall, New Jersey, 1988.

4. John Proakis, Digital Communications, McGraw-Hill, New York, 1995.

BIBLIOGRAPHY

1. Bernard Sklar, Digital Communications Fundamental and Applications, Prentice Hall, New Jersey, 1988.

2. John Proakis, Digital Communications, McGraw-Hill, New York, 1995.

3. I. A. Glover & P. M.Grant, Digital Communications, Prentice Hall Europe, 1998.

4. Communications Toolbox for Use with MATLAB and SIMULINK, The MATHWORKS Inc.

5. Hwei P.Hsu, Schaum's Outlines, Analog and Digital Communications, McGraw-Hill, 1993.

6. Dr. Jovan Lebaric, Notes for EC4550 (Digital Communications), Naval Postgraduate School, 1998 (unpublished).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Fort Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93940-5121	1
4.	Dr. Jovan Lebaric, Code EC/LB Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93940-5121	1
5.	Dr. Clark Robertson, Code EC/RC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
6.	Kara Kuvvetleri K.ligi MEBS. Bsk.ligi Bakanliklar, Ankara, Turkey	1
7.	Kara Kuvvetleri K.ligi Teknik ve Proje Yonetim Daire Bsk.ligi Bakanliklar, Ankara, Turkey	1
8.	Kara Kuvvetleri K.ligi Kutuphanesi Bakanliklar, Ankara, Turkey	1
9.	Kara Harp Okulu K.ligi Kutuphanesi Yenisehir, Ankara, Turkey	1
10.	Veysel Erdogan Ergenekon Mah. Kumkale Sok. No=27/9 Yenimahalle, Ankara, Turkey 06210	2